Technical Report 1165

Real Time Decision Alert, Aid and After Action Review System for Combat and Training

David S. Akin, Geoffrey E. Green, and Stephen J. Arntz
FSCX, Inc.

Larry L. Meliza
U.S. Army Research Institute

June 2005

**United States Army Research Institute
for the Behavioral And Social Sciences**

U.S. Army Research Institute
for the Behavioral and Social Sciences

A Directorate of the Department of the Army
Deputy Chief of Staff, G1

Authorized and approved for distribution:

MICHELLE SAMS	**ZITA M. Simutis**
Technical Director	**Director**

Research accomplished under contract
for the Department of the Army

FSCX, Inc.

Technical Review by

Jeffrey Stahl, RDECOM Simulation Training Technology Center
Billy L. Burnside, U.S. Army Research Institute

NOTICES

DISTRIBUTION: Primary distribution of this Technical Report has been made by ARI. Please address correspondence concerning distribution of reports to: U.S. Army Research Institute for the Behavioral and Social Sciences, Attn: DAPE-ARI-MS, 2511 Jefferson Davis Highway, Arlington, Virginia 22202-3926

FINAL DISPOSITION: This Technical Report may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: The findings in this Technical Report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

REPORT DOCUMENTATION PAGE

1. REPORT DATE (dd-mm-yy)	2. REPORT TYPE	3. DATES COVERED (from. . . to)
May 2005	Final	Jan 2003 – Feb 2005

4. TITLE AND SUBTITLE	5a. CONTRACT OR GRANT NUMBER
Real Time Decision Alert, Aid and After Action Review System for Combat and Training	DASW01-03-C-0011
	5b. PROGRAM ELEMENT NUMBER 665502
6. AUTHOR(S) David S. Akin, Geoffrey E. Green, and Stephen J. Arntz (FSCX, Inc.); Larry L. Meliza (U.S. Army Research Institute)	5c. PROJECT NUMBER M770
	5d. TASK NUMBER 234
	5e. WORK UNIT NUMBER C02
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) FSCX, Inc. 511 SW 'A' Avenue, Suite 2 Lawton, OK 73501-3927	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Institute for the Behavioral and Social Sciences, 2511 Jefferson Davis Highway, Arlington, Virginia 22202-3926.	10. MONITOR ACRONYM ARI
	11. MONITOR REPORT NUMBER Technical Report 1165

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

Contracting Officer's Representative and Subject Matter POC: Larry L. Meliza

14. ABSTRACT *(Maximum 200 words)*:

The System to Help Identify and Empower Leader Decisions (SHIELD) monitors command, control, communication, computers, and intelligence (C4I) data streams to alert leaders to situations requiring their attention (e.g., units violating a boundary). It allows leaders to temporarily dismiss alerts, have an alert go away for the rest of a mission, call up recommended courses of action, and/or call up job aids. It captures user responses to alerts in an interactive after action review (AAR) log file that can be used to host an AAR or the recipient of the alerts. SHIELD was designed to be used at any node within a C4I network while maintaining a small footprint. It has been demonstrated as a stand-alone system, as an application running on Force XXI Battle Command Brigade and Below (FBCB2) and the Command and Control Personal Computer (C2PC) without being integrated with these systems, and as an "injector" integrated with C2PC. Data collected by SHIELD to support AARs can also be used to support research on the placement of alerts within a network. Current efforts are directed towards implementing procedures to collect and analyze AAR logs across nodes to support unit level AARs and situational awareness research.

15. SUBJECT TERMS

Alerts After Action Review (AAR) During Action Review Force XXI Battle Command Brigade and Below (FBCB2) Command and Control PC (C2PC) Command, Control , Communications, Computers, and Intelligence (C4I)

SECURITY CLASSIFICATION OF			19. LIMITATION OF ABSTRACT	20. NUMBER OF PAGES	21. RESPONSIBLE PERSON
16. REPORT	17. ABSTRACT	18. THIS PAGE			
Unclassified	Unclassified	Unclassified	Unlimited	43	Ellen Kinzer Technical Publication Specialist (703) 602-8047

Standard Form 298

Technical Report 1165

Real Time Decision Alert, Aid and After Action Review System for Combat and Training

David S. Akin, Geoffrey E. Green and Stephen J. Arntz
FSCX, Inc.

Larry L. Meliza
U.S. Army Research Institute

Simulator Systems Research Unit
Stephen L. Goldberg, Chief

U.S. Army Research Institute for the Behavioral and Social Sciences
2511 Jefferson Davis Highway, Arlington, Virginia 22202-3926

June 2005

Army Project Number
665502M770

Small Business
Innovation Research

REAL TIME DECISION ALERT, AID AND AFTER ACTION REVIEW SYSTEM FOR COMBAT AND TRAINING

EXECUTIVE SUMMARY

Research Requirement:

The Division Capstone Exercise Final Report (2002) documented the warfighting capability of a force equipped with command, control, communications, computers, and intelligence (C4I) systems. The report recognized the explosive growth of data within the digitized Battlespace and concluded that: "the Army Battle Command Systems (ABCS) must have the ability to automatically track Commander's Critical Information Requirements (CCIR) and other criteria of success and provide automated alerts that will assist the decision maker in understanding the situation." The overall goal of this research was to demonstrate the capability and value of implementing a during action review system for digitized units that can be used in training as well as combat/operational environments.

Procedure:

The research requirement was addressed in the context of a Phase I and II Small Business Innovation Research (SBIR) effort funded by the Office of the Secretary of Defense. The feasibility and value of implementing a during action review system was demonstrated in developing a product called "System to Help Identify and Empower Leader Decisions (SHIELD)." SHIELD monitors digital (i.e., C4I) data streams, alerts the decision-maker to certain battle conditions requiring immediate action, and provides immediate access to recommendations and job aids. Two alerts were implemented within SHIELD; one alert is triggered when a friendly unit violates a boundary, and the second is triggered when a unit's plan for using artillery does not match what the intelligence section has determined to be enemy locations.

SHIELD development also considered the impact of alerts on the after action review (AAR) process. SHIELD was designed to capture alerts and leader responses to alerts in a format that can be used to host an AAR for recipients of alerts. SHIELD records whether the leader temporarily dismisses the alert, turns off a particular alert for the remainder of a mission (e.g., turns off a specific instance of a boundary violation), requests recommended courses of actions, and/or requests job aids. SHIELD also keeps a record of how many times a specific instance of an alert is triggered, and records whether the triggering event is addressed. To provide for the possibility that multiple instances of a particular alert type might occur within an exercise (e.g., multiple boundary violations) the capability was added for SHIELD to track the responses to each specific instance of a type of triggering event, separately.

SHIELD development was also conducted in a manner that provided information about where a during action review system might be located within a command and control network.

The intent was to implement SHIELD as a stand-alone system, then implement it as a non-integrated application running on select existing digital systems, and then implement it as an integrated application running on an existing digital system. To enable SHIELD to run as an application on existing digital systems without interfering with the operation of these systems, it was designed to have a small footprint in terms of random access memory (RAM) and processing requirements. The addition of a log file to support AARs set the stage for additional testing and refinement to ensure that the saving of alerts and leader responses to alerts would not impose heavy processing and memory loads on existing digital systems.

The software was designed to reduce the work required porting alerts from one location in the command and control network to another. The developmental approach also provided opportunities to port alerts and AAR log files developed at one node within a network to other nodes within the network.

Findings:

Two alerts and associated log files for AARs were implemented on a stand-alone system as might be employed by a battle captain in a tactical operations center (TOC). The alerts and log files were then ported to the Force XXI Battle Command Brigade and Below (FBCB2) system and the Command and Control PC (C2PC), in configurations where SHIELD was not integrated with the host system. This demonstrated the capability for SHIELD to be run as an application on an existing system, removing the need to bring yet another hardware platform into the digital network. At the request of the Project Manager for C2PC, SHIELD was integrated with C2PC by running as an "injector." The implementation of the AAR log file, in addition to providing a training feedback mechanism, also provided the capability for SHIELD to collect the data needed to decide whether placement of an alert at one node within a network is more effective than placement at other nodes.

The use of SHIELD as an embedded application on FBCB2 was found to have an impact upon FBCB2 functioning when FBCB2 was loaded on a Pentium II machine and engaged by a heavy processing requirement (i.e., calculating circular line-of-sight for distances of 12.5 kilometers). Less intensive calculations were not influenced by the operation of SHIELD.

Utilization and Dissemination of Findings:

SHIELD has the potential to serve as an operational tool, a training feedback tool, and as a research tool. At present, work is under way to add capabilities that will support all three of these applications. Four new alerts are being implemented, the capability for leaders or researchers to selectively turn off the display of specific alerts is being implemented, and the capability to collect and aggregate AAR log files across nodes and exercises is being implemented. Research to be conducted using SHIELD in the near term includes assessing the impacts of alerts on overall situational awareness.

REAL TIME DECISION ALERT, AID AND AFTER ACTION REVIEW SYSTEM FOR COMBAT AND TRAINING

CONTENTS

CONTENTS (continued)

List of Tables

List of Figures

REAL TIME DECISION ALERT, AID AND AFTER ACTION REVIEW SYSTEM FOR COMBAT AND TRAINING

INTRODUCTION

The After Action Review (AAR) Process

The U.S. Army's main method of providing feedback to units after collective training exercises is the after action review (AAR). The AAR is an interactive discussion in which units discuss what happened, why it happened, and how to improve or sustain performance in the future. This process can be facilitated and expedited through the use of AAR aids that illustrate key exercise events and/or show alternative ways to perform collective tasks (Morrison and Meliza, 1999). For example, the cause for a fratricide may become apparent by showing that friendly elements violated a boundary, entering another unit's sector.

The power of the AAR process, and of AAR aids, is based upon the capability to draw upon information from a variety of sources to provide an improved perspective regarding exercise events, resulting in greater awareness and understanding of the tactical situation after the fact. That is, the AAR process and aids may give the unit a view of the tactical situation that was not apparent to, or viewable by, any one exercise participant (Meliza, 1999). In many cases, the actions a unit decides to take in the future to correct performance may involve developing new tactics, techniques, and procedures (TTPs) that will improve a unit's awareness and understanding of the tactical situation *as it performs a mission.* For example, a unit may have attempted to synchronize activities between or among battlefield operating systems (e.g., maneuver and fire support) using time under the assumption that unit elements would be in a certain location at a specific time. Revised TTPs may call for synchronization to be accomplished using knowledge that a specific condition has been met (e.g., unit elements reporting that they had reached a specific location).

The Potential for During Action Review (DAR) Aids

The U.S. Army demonstrated the capability for software to automatically generate AAR aids during exercises for immediate use at the end of an exercise (Brown et al., 1997). This demonstration was conducted in the virtual, networked simulator training environment known as SIMNET. The SIMNET environment provided a stream of simulation data that could be analyzed by software to decide when certain key events occurred (e.g., when a unit crossed a particular phase line), and then the software could create a particular type of aid capturing a potentially important aspect of unit performance (e.g., a snapshot of the position of individual vehicles showing the type of movement formation being used at a specific point in time). Certain of the AAR aids automatically generated, if provided to units in mid exercise as a during action review (DAR) aid might cue units to address a performance problem in time to influence mission outcome (e.g., directing a subordinate leader to adopt a more appropriate formation). Providing such information risks the possibility of training units to depend upon cues that would not be available in the operational environment. This situation changed with the advent of command, control, communications, computer, and intelligence (C4I) systems, because the

operational C4I stream may replace the simulation data stream as a source of data for automatically generating both DAR and AAR aids.

In terms of the current digital force, this C4I data stream includes the Force XXI Battle Command Brigade and Below (FBCB2) system on board tactical platforms and the Army Battle Command Systems (ABCS) employed by battle staffs in tactical operations centers (TOCs). FBCB2 provides the digital network with data on the location of FBCB2-equipped platforms, facilitates transmission of graphical and textual data among units and between units and staffs, and provides leaders with analytical tools to enhance situational awareness (SA). C4I systems in the TOC support the performance of section-specific functions and facilitate the sharing of evolving planning products among sections and echelons and with FBCB2-equipped units.

C4I is expected to provide improved perspectives on exercise events, SA, and situational understanding in its own right; however, C4I systems do not appear to completely meet the need for a DAR capability. The Division Capstone Exercise Final Report (2002) documented the warfighting capability of a digitized force operating in a contemporary operational environment (COE). The report recognized the explosive growth of data within the digitized Battlespace and concluded that C4I systems must have "the ability to automatically track Commander's Critical Information Requirements (CCIR) and other criteria of success and provide automated alerts that will assist the decision maker in understanding the situation." The need for alerts may be especially critical for FBCB2-equipped units, because unit leaders and vehicle commanders cannot be expected to continually monitor their FBCB2 SA displays. In laboratory situations it has been shown that ability of individuals to track even limited portions of the tactical situation using SA displays is severely reduced when individuals are performing other tactical activities (Durlach and Chen, 2003). The implementation of alerting mechanisms is one possible solution to this problem (Durlach and Meliza, 2004).

Exploring the DAR Concept through the Development of a Prototype "System to Help Implement and Empower Leader Decisions (SHIELD)"

The U.S. Army Research Institute (ARI), under the sponsorship of the U.S. Army Training and Doctrine Command (TRADOC), initiated a Science and Technology Objective (STO) called "Methods and Measures of Commander-Centric Training" to help guide the development of training products and procedures appropriate to a C4I enabled force. ARI envisioned the need for a software product that might be used to monitor the digital data stream, create DAR aids, and then display these aids to units in time for units to take appropriate actions. ARI also envisioned the need for a testbed that could be employed in addressing a variety of research/development issues regarding the use of a DAR capability. The Office of the Secretary of Defense agreed with these visions and funded a Small Business Innovation Research (SBIR) project called "Realtime Collective Feedback for Combat." This report documents the lessons learned from development of the System to Help Implement and Empower Leader Decisions (SHIELD) during Phases I and II of the subject SBIR project. The main thrust of this effort was to demonstrate the feasibility and value of a DAR system. Issues addressed in this report include:

- What are the features of a DAR mechanism that can be responsive to the needs of decision-makers without becoming a distraction?
- How does the DAR process differ from the AAR process?

- Where within a network should a DAR capability be placed, and what is the impact of placement on design features?
- What are the cost and benefit issues associated with tapping the various types of C4I data streams?
- What is the impact of having DAR aids on the AAR process?
- Are the DAR aid capabilities limited to alerting functions?
- What are the features of a testbed that can support research on the application of DAR aids?

Organization of Report

The first section of the report describes DAR aids in the form of alerts, recommended courses of action, job aids, and intrusiveness controls built into SHIELD. The second section addresses placement of DAR aids within a network, in terms of decision nodes (i.e., leadership roles) and the degree to which a DAR system is embedded within a tactical network (i.e., a stand alone system, an application running on the same platform as an existing C4I system or systems, or an application integrated with one or more existing C4I systems.) The third section focuses on the impact of a DAR process and system on the AAR process and AAR system. The fourth section focuses on the C4I data streams that can be tapped to create DAR aids. The fifth section describes a testbed developed to address DAR application and SA research issues.

IMPLEMENTATION OF SAMPLE DAR AIDS

Table 1 describes the challenges faced by C4I-enabled units and potential solutions offered by SHIELD. Currently, commanders and staffs must monitor and painstakingly analyze a daunting amount of digital information during the heat of battle planning, preparation, and execution to discern what's truly relevant and critical to their mission. As the Future Force and network centric operations evolve, the situational understanding challenge increases. To implement the key tenets of the Future Force--"see first, understand first, act first and finish decisively", decision-makers must have modern decision support tools that can assist them in understanding and acting on critical battlefield situations in a timely manner.

Table 1. Digital Command and Control System Battlefield Challenges and SHIELD Solutions

Digitization Challenge	SHIELD Solution
Potential for Information Overload	Alerts for User Defined Key Events
Skipped/forgotten steps or processes as a result of operator stress, fatigue, lack of experience, changing situations on the battlefield, and/or changed or new SOPS/Battle Drills	Automated Alerts, Recommendations and Job Aids

Digitization Challenge	SHIELD Solution
Potential problems with the operator's data synthesis	Conversion of data to knowledge. SHIELD presents critical events in such a way that operators can quickly and easily understand what the situation is and what they need to do about it. SHIELD generates enhanced situational awareness displays using data from multiple digital systems and provides information to the user that is not readily available in his or her current digital system.

Implementations of SHIELD Alerts, Recommended Courses of Action, and Job Aids

The initial plans for implementing the SHIELD concept called for two rule sets to be applied. One rule set was concerned with deciding whether friendly units violated a boundary, thus risking the possibility of being mistaken for enemy elements by the friendly units whose sector had been entered. The second rule set was concerned with deciding whether there is a mismatch between the location of planned fire support missions and current awareness regarding the location of enemy forces. Violations of the first rule set produced the Fratricide Prevention/Cross Boundary Violation alert, and violations of the second produced the Fire Plan Update alert.

The overall concept for the SHIELD alerting mechanism is illustrated in Figure 1. SHIELD parses the C4I data stream to update specific facts about the tactical situation. United States Message Text Format (USMTF) and Joint Variable Message Text Format (JVMF) messages provide SA data used by SHIELD to write rules and trigger alerts. When SHIELD receives one of the SA messages it is monitoring, it updates its own database of SA facts. It then sends these facts through rule sets to determine whether the information violates the rules. If a SA message does not violate the rules, then SHIELD does nothing but update its information. If the SA message violates the rules, then SHIELD alerts decision-makers.

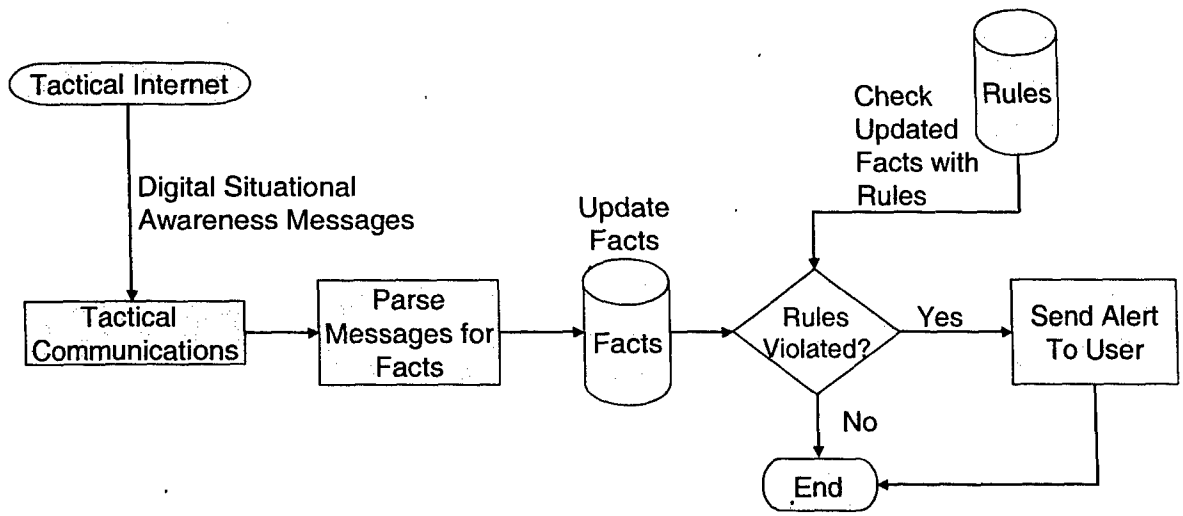


Figure 1. Concept for SHIELD alerting mechanism.

For the Boundary Violation alert, SHIELD discerns from the JVMF message traffic which section of the map constitutes the boundaries for the Task Force and which units belong to the Task Force. These three JVMF messages establish the facts needed to run SHIELD's expert rule for the "Fratricide Prevention/Cross Boundary Violation" task. These messages provide the information SHIELD needs to determine which units are authorized to be in the Task Force's area of operations, and prompts SHIELD to alert decision-makers when an unauthorized unit crosses into the Task Force's area of operations or when one of the Task Force's units goes outside its area of operations. SHIELD includes an alert package that draws a topographical map, icons, and an area of operation onto an image, and then encodes that image into an animated graphics interchange format (GIF) file. The alert displays text identifying the violating entity and where it was located at the same time that the "alert map" displayed the violating icon as a flashing icon in the center of the alert display. Figure 2 illustrates a boundary violation alert and decision-maker options supported by the graphical user interface (GUI).

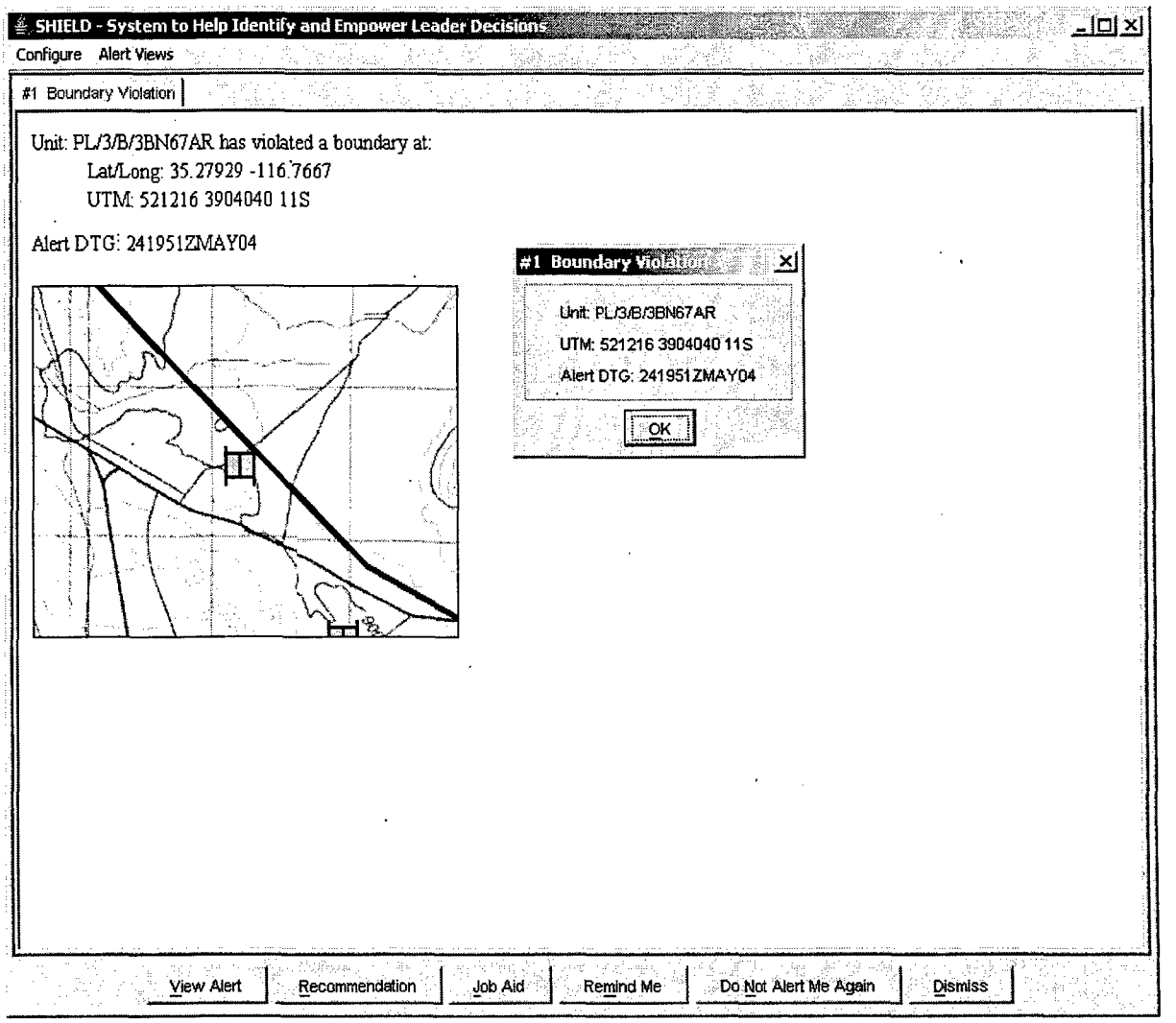


Figure 2. SHIELD boundary violation alert.

To fully appreciate the benefits of an alert for boundary violations, one must consider what the FBCB2 SA display is likely to look like for higher echelons. To avoid cluttering the screen with friendly icons, leaders at battalion and above are likely to set their systems to display aggregate icons showing the center of mass of units rather than showing the location of each individual platform. Such a display is unlikely to reveal to a leader that a boundary violation has occurred, unless, for example, an entire platoon violates the boundary and a leader is looking at a display that aggregates at platoon level. SHIELD, on the other hand, looks at the location of individual vehicles, and SHIELD provides very specific information regarding a boundary violation (i.e., specific vehicles and locations) rather than alerting a leader to a violation and leaving it up to the leader to find out the specifics.

There may be cases where a leader would like to have access to recommended courses of action or even job aids, and SHIELD was designed to meet this need for guidance. Figures 3 and 4 show the recommended courses of action and job aids, respectively, associated with the Boundary Violation alert.

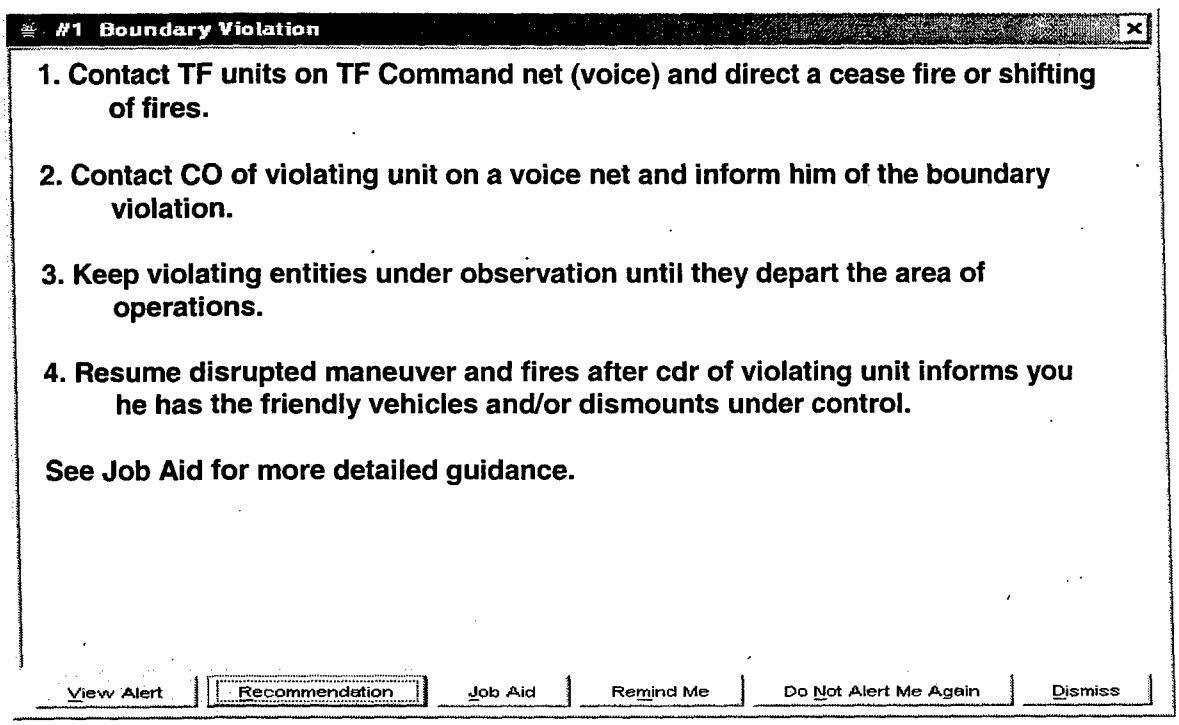


Figure 3. SHIELD recommended courses of action for a boundary violation.

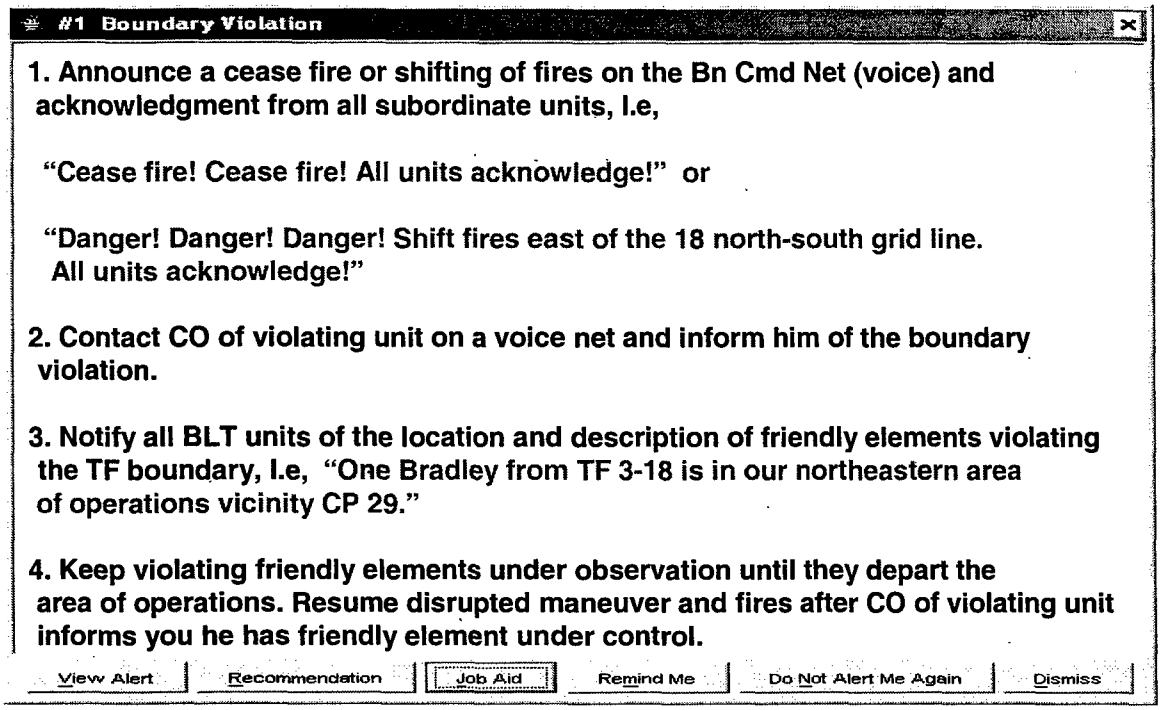


Figure 4. SHIELD job aid for a boundary violation alert.

Figure 5 shows the second alert implemented within SHIELD. This aid was created using the C4I data stream available to the Advanced Field Artillery Tactical Data System (AFATDS). The rule set triggering this aid checks to see whether there are enemy positions that have not been targeted for artillery missions. The rule also checks to see whether there are locations targeted for artillery missions that are not currently associated with enemy locations. Changes in fire plans or changes in known enemy locations can trigger the application of the rule. Information about suspected enemy locations are taken from an All Source Analysis System (ASAS) USMTF message. The location of artillery targets is from the AFATDS Target List, and this information is extracted from a JVMF message. Recommended courses of actions and job aids were also developed for this second alert.

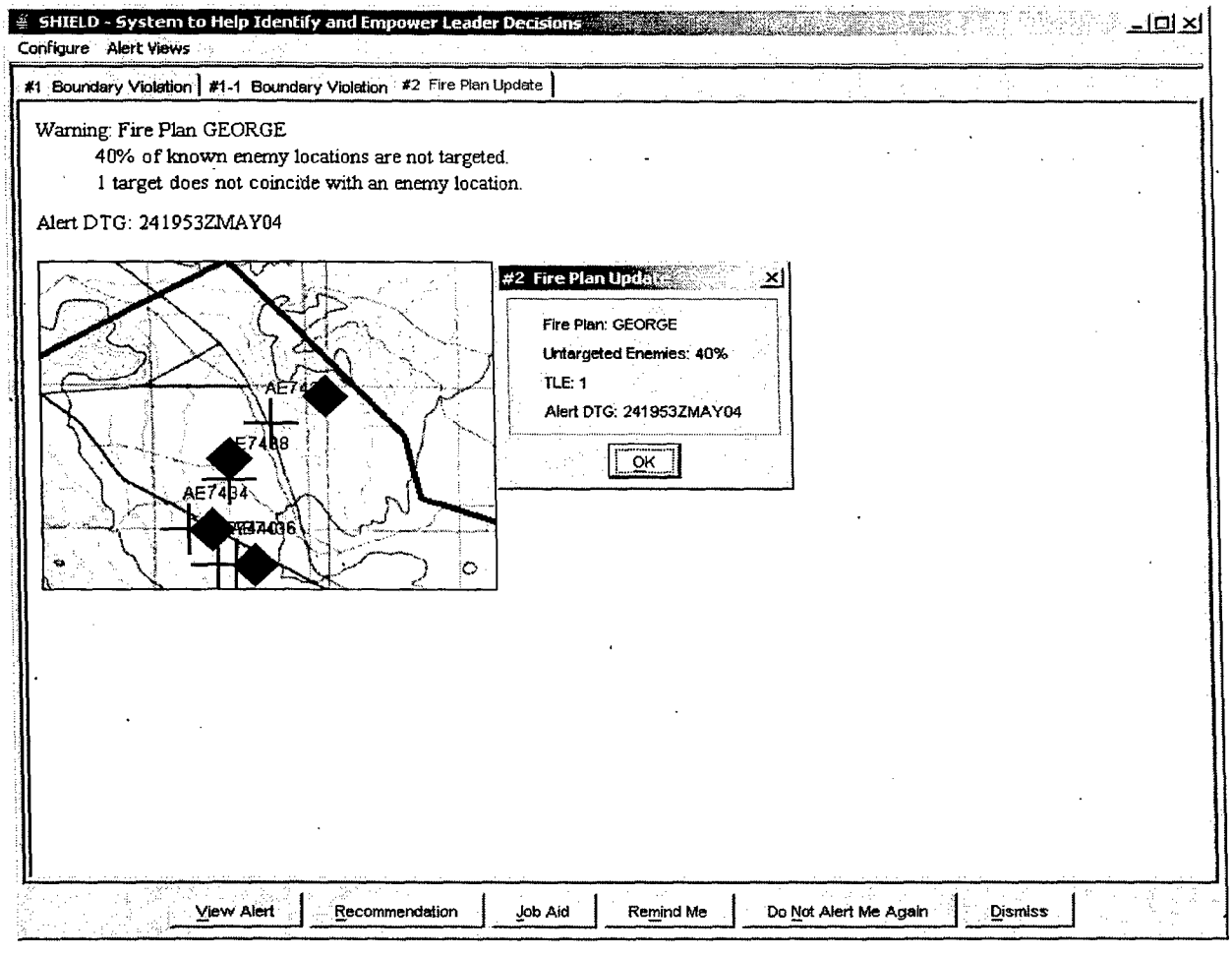


Figure 5. Fire Plan Update alert.

Controlling Intrusiveness of Alerts

SHIELD's alert triggering mechanisms (expert rules) are designed to recognize a problem early, giving leaders the opportunity to take action before the battlefield situation impacts adversely on the unit mission or force protection. On the other hand, alerts can be intrusive, interrupting combat tasks that are already being performed by leaders. For this reason it is important for leaders to be able to control the intrusiveness of alerts. In the case of SHIELD alerts, a leader may simply dismiss the alert to get if off the screen. In this case, the alert will reappear the next time the SHIELD rule set checks the situation and finds out a rule violation still exists. SHIELD also offers leaders the options of having SHIELD repeat the alert at a later time or instructing SHIELD not to repeat the alert. If a leader selects the option of not repeating the alert, SHIELD continues to check whether the rule violation exists and creates and logs the alerts, but the alerts will not be displayed automatically. Instead, the recorded alerts will be retained in a log file for user-initiated reviews at the end of an exercise or mission. This feature will be described further under the section of this report that addresses AAR capabilities.

SHIELD's degree of intrusiveness was 'hard-coded for this Phase II SBIR," except that leaders were given the ability to dismiss alerts from the screen, have the alert be repeated at what may prove to be a more convenient time, or turn off the alert. Figures 6, 7, and 8 illustrate GUIs

that might be implemented to provide leaders the ability to tailor alerts by configuring the intrusiveness filters prior to training or combat operations. Configuring SHIELD includes setting geographical and operational geometry and selecting intrusiveness settings that incrementally reduce SHIELD interventions. Intrusiveness filter settings identify conditions when the leader does not want alerts displayed, e.g., if within [n] meters of a known threat location, if [n] alerts are already displayed, alerts are not actioned after [n] period of time, etc. [N] represents the leader's ability to quantify the level of intrusiveness. Regardless of the intrusiveness filter settings selected, SHIELD continues to monitor and record all alerts. Units may establish default settings by unit standard operating procedures (SOP) so that leaders will not need to modify filter settings for each training event or tactical operation.

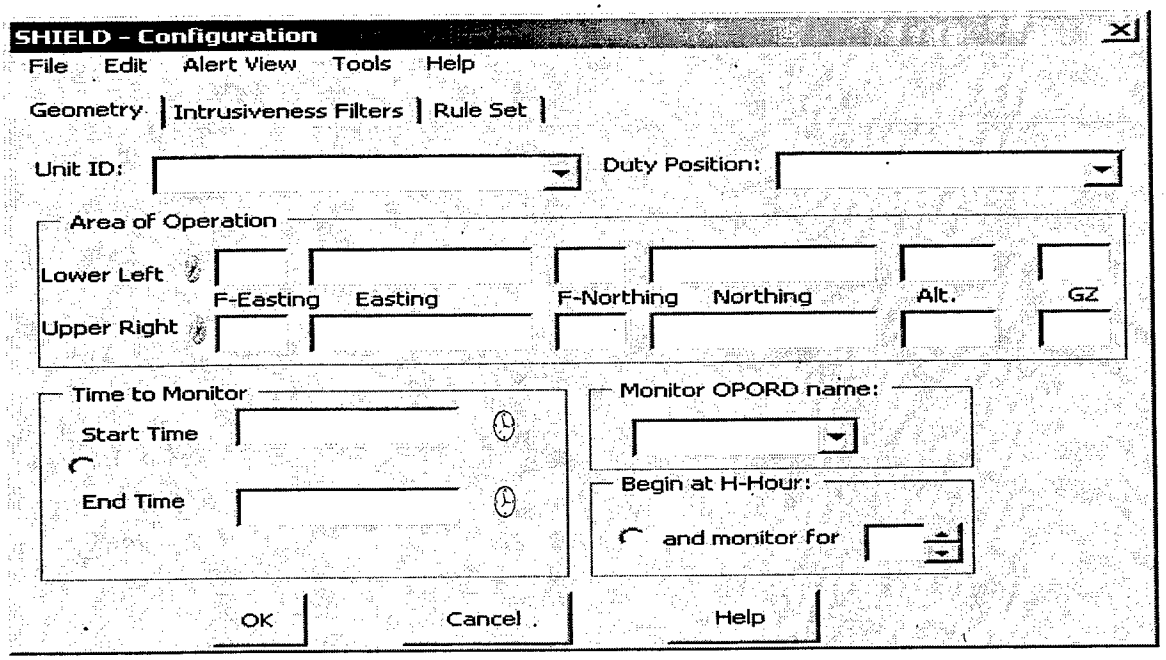


Figure 6. SHIELD geometry filter.

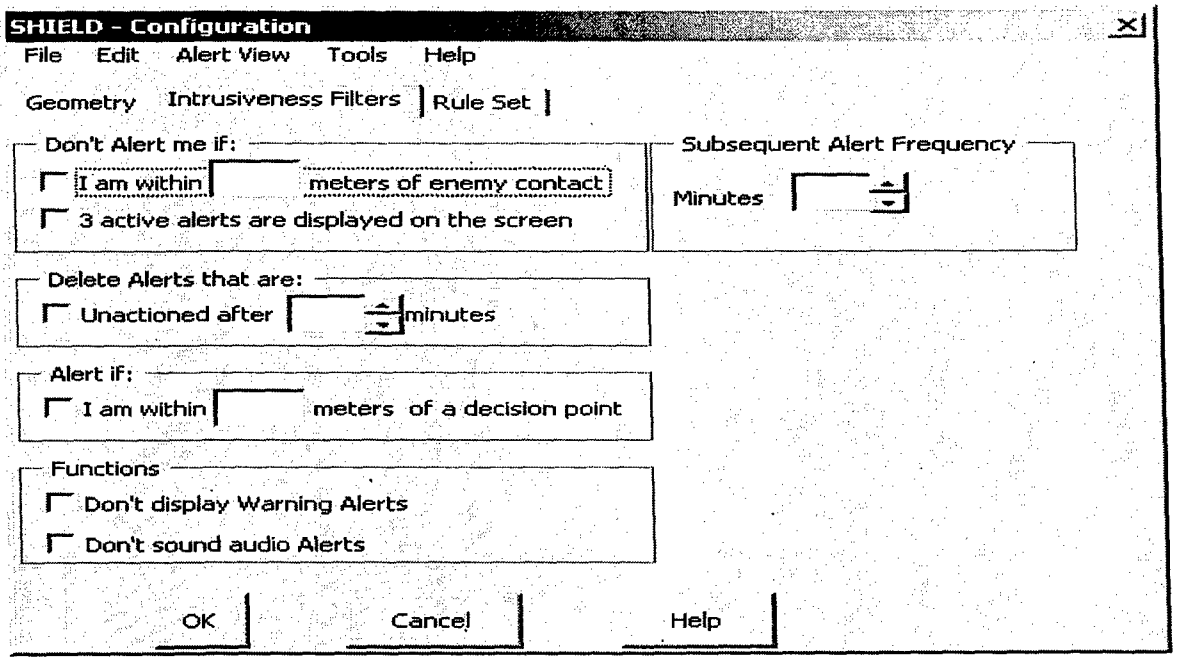


Figure 7. SHIELD intrusiveness filter.

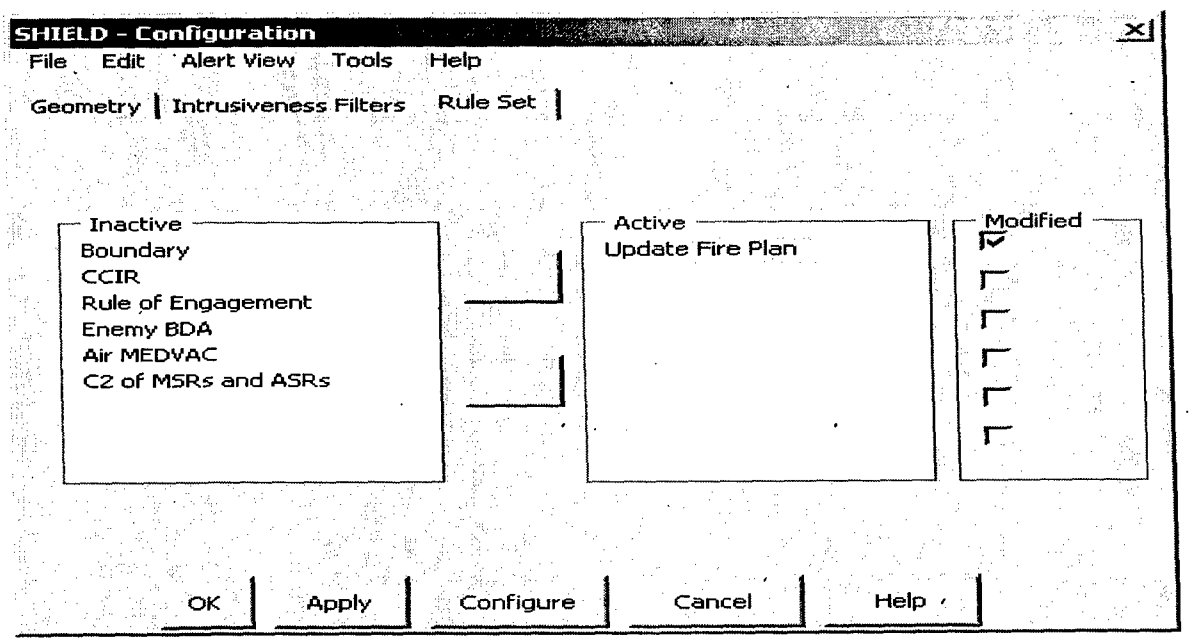


Figure 8. SHIELD rule set filter.

To date, none of these filters have been implemented; however, FSCX is implementing a filter that will allow users and researchers to selectively turn the display of specific types of alerts on or off prior to the start of an exercise. This capability is being implemented in a way that will cause rules to be applied and alerts to be recorded for future use, even when the display of the alerts has been turned off. Four new types of alerts are being implemented as part of this modification of SHIELD.

DECIDING WHERE TO PLACE A DAR SYSTEM WITHIN A NETWORK

SHIELD was developed in an iterative fashion with two broad goals in mind. One goal was to progress as far as possible in terms of actually integrating SHIELD with an existing C4I system. This goal comes from the fact that, in many tactical environments, there is no more room for additional stand-alone systems. The second goal was to develop information about the ease or difficulty of porting SHIELD or specific SHIELD alerts from one point in a C4I network to another, such as transitioning from a stand-alone system to a system embedded within a C4I system and/or porting from one C4I system to another. This goal is important because the optimum location of specific alerts within a network may not be known. In addition, changes in C4I systems may influence the optimum location of specific alerts.

The plan to reuse SHIELD at multiple nodes within a network required software approaches that enhance the potential for reuse. The plan to have SHIELD run on the same platform as a C4I system required designing software capable of maintaining a small footprint, in terms of random access memory requirements, to avoid interfering with the performance of the host system.

Iterative Design of SHIELD Versions Varying in Terms of C4I System Integration

In Prototype 1, SHIELD was completely independent of any C4I system, in that it used its own computer and screen. Prototype I was described as a stand-alone version. This version made it possible to monitor the C4I data stream that flows along the local area network (LAN) in the TOC and use the system's own hardware and software to display SHIELD alerts, enhanced SA displays, recommended courses of action, and job aids. In concept, a number of stand-alone SHIELDS might be available in a given TOC, each responsible for a unique set of alerts. Realistically, TOCs already have a large number of C4I platforms and there is not enough room to add multiple stand-alone versions of SHIELD.

The targeted decision-maker for a stand-alone is the TOC Battle Captain, who can use SHIELD to promote the integration of battlefield operating systems (BOS) during the planning and execution of tactical operations. For example, one of the SHIELD rules alerts could alert the Battle Captain when planned fire support targets did not coincide with enemy positions. SHIELD provided a brief text-based, graphical alert stating the percentage of planned fires that would be ineffective. Additionally, SHIELD provided an enhanced SA display, showing the planned targets in relation to the enemy positions. The Battle Captain could then direct the TOC Fire Support Element to shift planned fires to known enemy locations revealed by late-breaking intelligence. This prototype successfully demonstrated SHIELD's real-time feedback proof of concept.

In Prototype 2, the SHIELD software developed for Prototype 1 was installed on an FBCB2 surrogate PC. There was no interface between the two applications running on the same computer. SHIELD monitored JVMF and USMTF digital messages that it received from the TOC LAN/Simulated Tactical Internet Connection, as re-created in the FSCX Lab. As the simulation displayed the situation on the FBCB2 platform, SHIELD provided its alerts and access to recommended courses of action and job aids on the same FBCB2 platform and in the same format as on the stand-alone version. SHIELD was also demonstrated running as an

application on the Command and Control PC (C2PC). C2PC is a U.S. Marine Corps Systems Command C4I system, used in U.S. Army TOCs at brigade level to provide a joint view of the tactical situation. Prototype 2 demonstrated SHIELD's real-time feedback proof of concept without need for additional PCs/Computer sets. This configuration provides for the possibility of having multiple instances of SHIELD within a TOC without adding more hardware platforms. Similarly, this configuration provides for the possibility of SHIELD applications running on individual FBCB2-equipped tactical vehicles without adding more hardware. Under this configuration, each instance of a SHIELD could be creating different sets of alerts. For example, boundary alerts might be employed on the Maneuver Control System (MCS) used by the S3 in the Battalion TOC and on the FBCB2 systems of company commanders. The Fire Plan Update alert might be provided on the AFATDS.

Given the relevance of the Fire Plan Update alert to the fire support BOS, FSCX planned to test this rule running as an application on AFATDS; however, the AFATDS Product Manager would not authorize FSCX to have access to the AFATDS codes that would enable SHIELD to run on the same platform as AFATDS. As a result, FSCX used SHIELD on a stand-alone platform to test its ability to monitor the same message traffic sent to AFATDS and to provide alerts to events triggering both the "Cross Boundary Violation/Fratricide Prevention" and the "Fire Plan Update" rules. As a point of clarification, AFATDS receives information about enemy locations from a USMTF message, while FBCB2 receives this information from a JVMF message.

In Prototype 3, SHIELD was integrated with FBCB2 only to the point that it included a SHIELD TAB on the FBCB2 GUI used to activate SHIELD. Otherwise, SHIELD continued to operate the same as Prototype 2 (i.e., a separate software program running on the same FBCB2 platform). As part of the development of Prototype 3, FSCX also determined the interfaces needed to access the C4I Device's Data Base, focusing on messages received and sent, plans developed on the C4I, but not sent; and Line of Site tools and routing, that would be beneficial for SHIELD to use in future versions. FSCX also identified requirements and feasibility of total integration of SHIELD with ABCS software for evaluation and application in Prototype 4.

In Prototype 4, FSCX fully integrated SHIELD with existing C2PC software. This allows SHIELD to access tools, maps, and other information not available through monitoring the digital data stream. This integration was made possible by the fact that the Project Manager for C2PC encourages the development of third party software to enhance the performance of C2PC. At the request of PM C2PC, SHIELD was integrated with C2PC as an "injector" to support a demonstration for the C2PC Joint Configuration Control Board (JCCB). Figure 9 illustrates SHIELD running as an "injector" within C2PC.

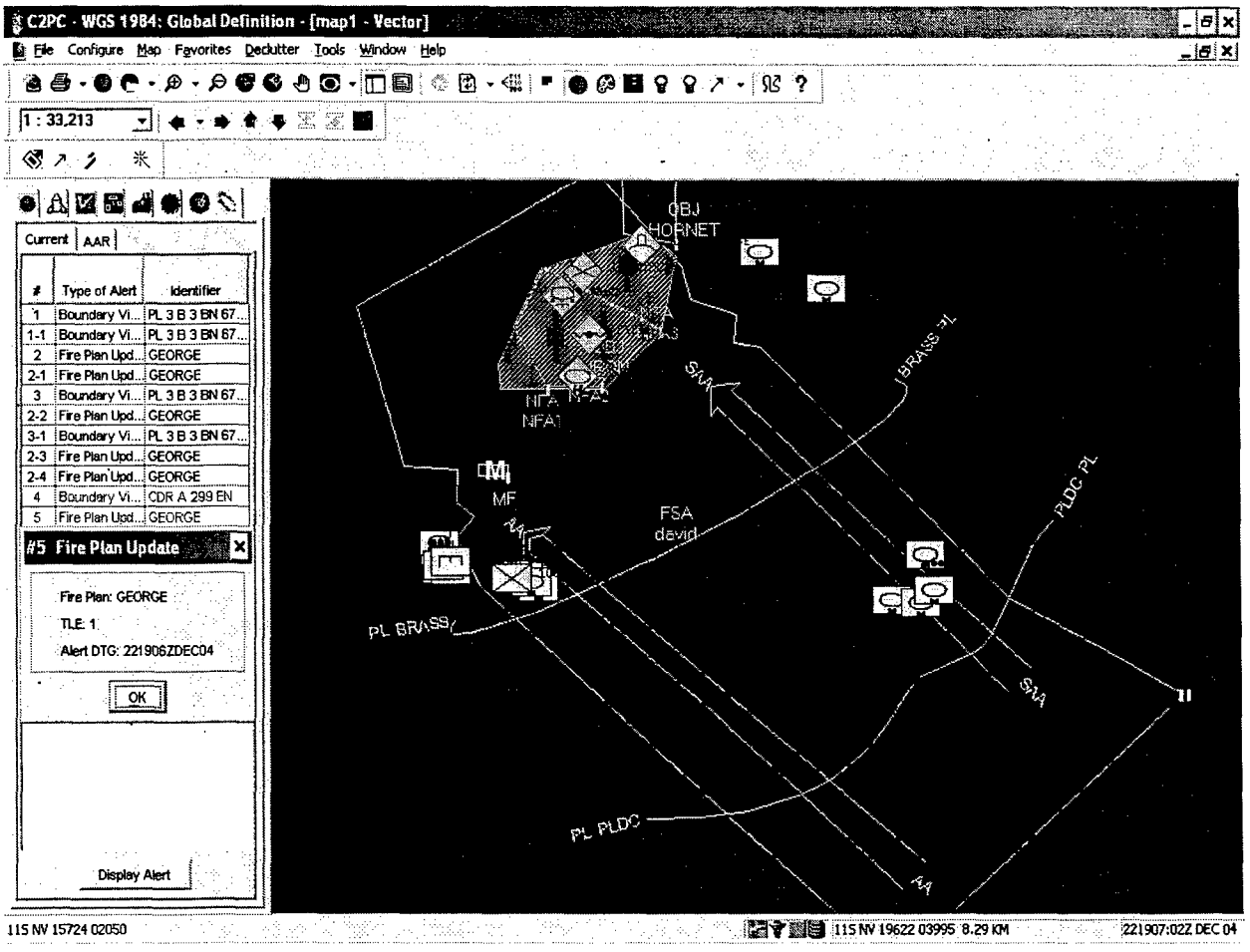


Figure 9. SHIELD – injected on C2PC.

The results achieved above made it clear that installing SHIELD on an ABCS/C4I system is not only technically feasible, but also makes the most sense, particularly in terms of not adding another platform into the TOC or to FBCB2-equipped platforms. However, either programmatic or proprietary issues, or both, may dictate that a stand-alone SHIELD will be the only viable solution for those specific ABCS/C4I platforms on which it is not allowed to be installed. Of course, given its small footprint, completely embedding SHIELD with an ABCS/C4I system would be best from a technical perspective--particularly for emerging and future systems. In summary, SHIELD can operate embedded on a C4I system, installed on a C4I system or operated in a stand-alone configuration, depending on programmatic constraints rather than technical constraints.

Designing SHIELD to be Reused Across Nodes

A variety of tactics were used to make it easy to reuse SHIELD across nodes in a network. The purpose of any software design is to implement requirements -- providing the "how" to accomplish the "what." Even a cursory review of the specifications developed during Phase II indicated it would be necessary to interface with many ABC systems, and develop a significant number of rules, alerts, recommended courses of action, and job aids. Accordingly, the two most important design considerations were to develop SHIELD so it could: (1) readily

interface with multiple ABC systems; (2) execute rules, alerts, enhanced SA displays, recommended courses of action, and job aids in the same manner regardless of the C4I system. This section describes how these design priorities were achieved.

The key to building any software system independent of a subsystem is to *hide* the implementation details of the subsystem. SHIELD was developed so that it *doesn't* know any details about ABCS systems, or at least as few and as generic as possible. To do this, FSCX developed an interface, called the PlatformFacade. The PlatformFacade is an example of the Façade pattern [ref. Design Patterns] that "provide[s] a unified interface to a set of interfaces in a subsystem (Gamma, et al, 1995). Façade defines a higher-level interface that makes the subsystem easier to use."

While this approach would help to hide the details of any one ABCS system– it was necessary to extend the approach, so that all ABCS systems could be hidden behind a generic interface. To do this, FSCX capitalized on the core component of SHIELD—its rule engine. As an expert system, SHIELD contains an engine that makes decisions based on available data and rules provided to the engine. Data provided to the engine come from SHIELD's interface with the ABCS platform. To make the PlatformFacade generic, FSCX developed a generic interface (the Façade) to the ABC system, and other C4I systems, that defines how data is supplied to the rule engine. The rule engine (Java Expert System Shell [JESS] to the C Language Integrated Production System [CLIPS]) receives and acts on data in the form of Fact objects. As such, the PlatformFacade defines the method of supplying data to the rule engine by asserting Fact objects. Once this is done, the PlatformFacade loads an interface to the C4I system, which then asserts standardized Facts to SHIELD. In turn, the rule engine uses the Facts to determine if an operator alert, recommended course of action, and job aid are necessary. One major benefit of the SHIELD design is that it hides unique C4I system information, such as USMTF/JVMF message formats and databases from SHIELD, which are largely irrelevant to the SHIELD code itself. Thus, message formats or databases can change and new C4I systems can emerge and SHIELD will remain relevant. Another benefit is that it facilitates developing recommended courses of action and job aids, since all the facts available are developed when the interface to the C4I system is developed.

Since SHIELD's design permits interfacing with multiple C4I systems, and since the facts available from a particular C4I system implementation are known, it would be convenient to create a large number of rules and add them to SHIELD without having to change any source code. To do this, recommended courses of action and job aids were treated as *data;* not source code. JESS rule engine already accepts rules in American Standard Code for Information Exchange (ASCII) text format. Alerts, recommended courses of action, and job aids were defined in a way that allows SHIELD to read, manipulate, and display them at run-time. To do this, two relatively new Internet standards were applied: eXtensible Markup Language (XML) and eXtensible Stylesheet Language Transform (XSLT). XML is a method for defining data without associating any representation information. In other words, how the data is displayed is not coupled to the data itself. XML can be generated and parsed dynamically, and there are numerous packages and Application Programming Interfaces (API) in all of the major programming languages. In addition, XML is defined by a *document type definition* appropriate

for the application using the XML. FSCX prepared an XML definition for recommended courses of action and job aids; other applications can use the definition for the same data.

XSLT uses stylesheets and an engine to transform generic XML data into a particular graphical representation. For example, XSLTs exist to transform XML into HTML, for use in web browsers. The XSLT stylesheets are also written in XML. For SHIELD, the rule engine uses rules to generate dynamic recommended courses of action and job aids in XML, which are then converted into HTML for display. This powerful approach to dynamically generate information for the user is all based on standards, and allows for adding numerous rule sets and associate recommended courses of action and job aids to SHIELD functionality, without modifying SHIELD's source code.

When SHIELD is employed as an application on a digital system, it may employ any digital data that come to the node where it is employed, regardless of whether the host C4I platforms use those data. This is one of the methods by which SHIELD can offer an improved perspective on the tactical situation relative to that provided by existing C4I systems.

Designing SHIELD to Reduce the Possibility of Interfering with C4I systems

An important variable relevant to SHIELD placement is that of avoiding interference with C4I systems. If a single instance of SHIELD attempts to monitor too many aspects of the tactical situation, it will consume too much processing power and risk interfering with C4I functionality. If SHIELD is distributed across a variety of C4I systems and echelons, and each instance of SHIELD produces a small, select variety of alerts, then the possibility that SHIELD will interfere with C4I systems is reduced. It is possible to move specific alerts from one node to another to ensure that alerts are provided at the most effective location.

SHIELD was designed in a way to reduce potential impacts on C4I systems. SHIELD *per se* requires only three megabytes of RAM. One test of SHIELD impacts on operational C4I systems was to look at the impact of running SHIELD on the time required for the FBCB2 line-of-sight (LOS) and circular LOS (CLOS) functions to complete their calculations and display their results for distances of one, six, and 12.5 kilometers. This data collection was conducted in a situation where SHIELD was analyzing an exercise data stream of approximately 30 C4I messages/second, and the FBCB2/SHIELD combination was running on a Pentium II platform with 192MB RAM at 450 Mhz. Table 2 provides results of processing times for the CLOS application only; LOS calculations and results at all ranges and configurations were perceived to be instantaneous with and without SHIELD operating (=/< 1sec.). No degradation was observed for CLOS for distances of six kilometers and below.

Table 2. Effect of SHIELD on FBCB2 CLOS Performance

Platform Configuration	1,000 Meters	6,000 Meters	12,500 Meters	
SHIELD embedded on FBCB2. Simulation not running	1 sec	3 – 5 sec	7 - 8 sec	Process Time
SHIELD embedded on FBCB2. Simulation not running	1 sec	3 – 5 sec	7 - 8 sec	Process Time
SHIELD embedded on FBCB2. Simulation not running	1 sec	3 – 5 sec	8 - 9 sec	Process Time
SHIELD embedded on FBCB2. Simulation running	1 sec	3 – 5 sec	10-12 sec	Process Time

An important variable influencing SHIELD memory requirements is the amount of memory required to save alerts and their associated enhanced SA displays, because the SA displays contain graphical data. To further reduce memory requirements, SHIELD was modified so that it would save definitions of alerts rather than alerts *per se*. Saving the definition of the situation allows alerts to be created on demand without having to save large graphic files containing map displays. Additional memory benefits were gained by integrating the C4I message parser with SHIELD. The benefits of these memory reduction efforts were assessed in terms of their impacts on stress tests in which large numbers of alerts were generated until SHIELD was no longer capable of creating new aids. In the case of running SHIELD on FBCB2, eighty-five alerts were generated before the memory reduction effort, some with flaws. After the memory requirement was reduced, 785 alerts were generated on the FBCB2/SHIELD combination, without any errors. In the case of C2PC, stress testing after the memory reduction intervention was terminated after over 4,300 alerts had been produced. For both C4I systems, the number of alerts produced after the intervention is far beyond the number of alerts likely to be produced during an exercise or mission.

IMPACT OF DAR ON THE AAR PROCESS

SHIELD takes some of the information that used to be provided to units during AARs and presents it during missions or exercises, in time for units to make use of this information to influence mission/exercise outcomes. In addition, how decision-makers respond to alerts becomes a new topic to be addressed by AARs. FSCX implemented the capability to collect the alerts, enhanced SA displays, and information regarding user responses to alerts for use during AARs. The information from the AARs may also be used for long-term training in terms of identifying trends and training deficiencies. It will also provide opportunities for leader input to SHIELD rule refinement and new rule development processes. Although implementation of the AAR capability was not required by the Phase II SBIR contract, it was a logical extension of SHIELD's functionality that served to further demonstrate the value of a system that filters the C4I traffic to alert decision-makers to situations requiring their attention. Portions of the SHIELD mechanism may even be used to create and capture AAR aids that are not related to alerts, such as conditions deemed prior to mission execution as not critical enough to be alerts but still useful enough for AAR purposes. Figure 10 attempts to provide a complete picture of how SHIELD can fit into the training process.

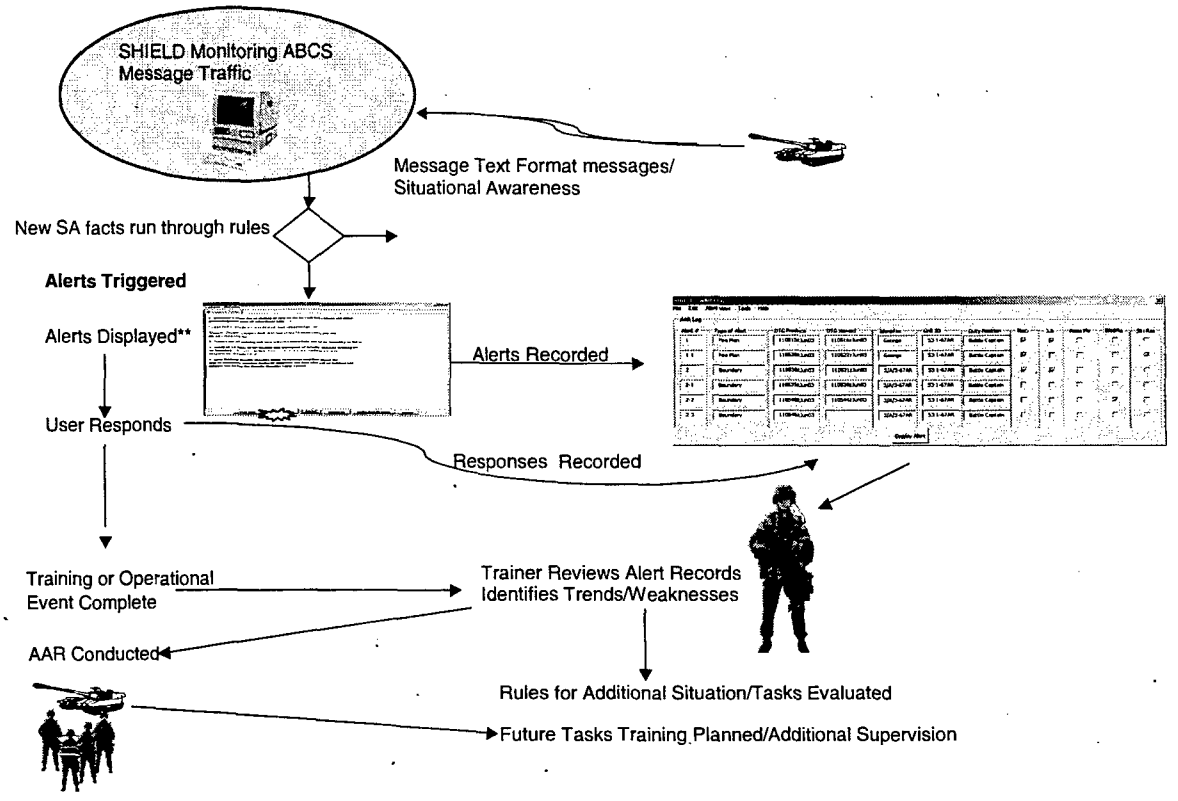
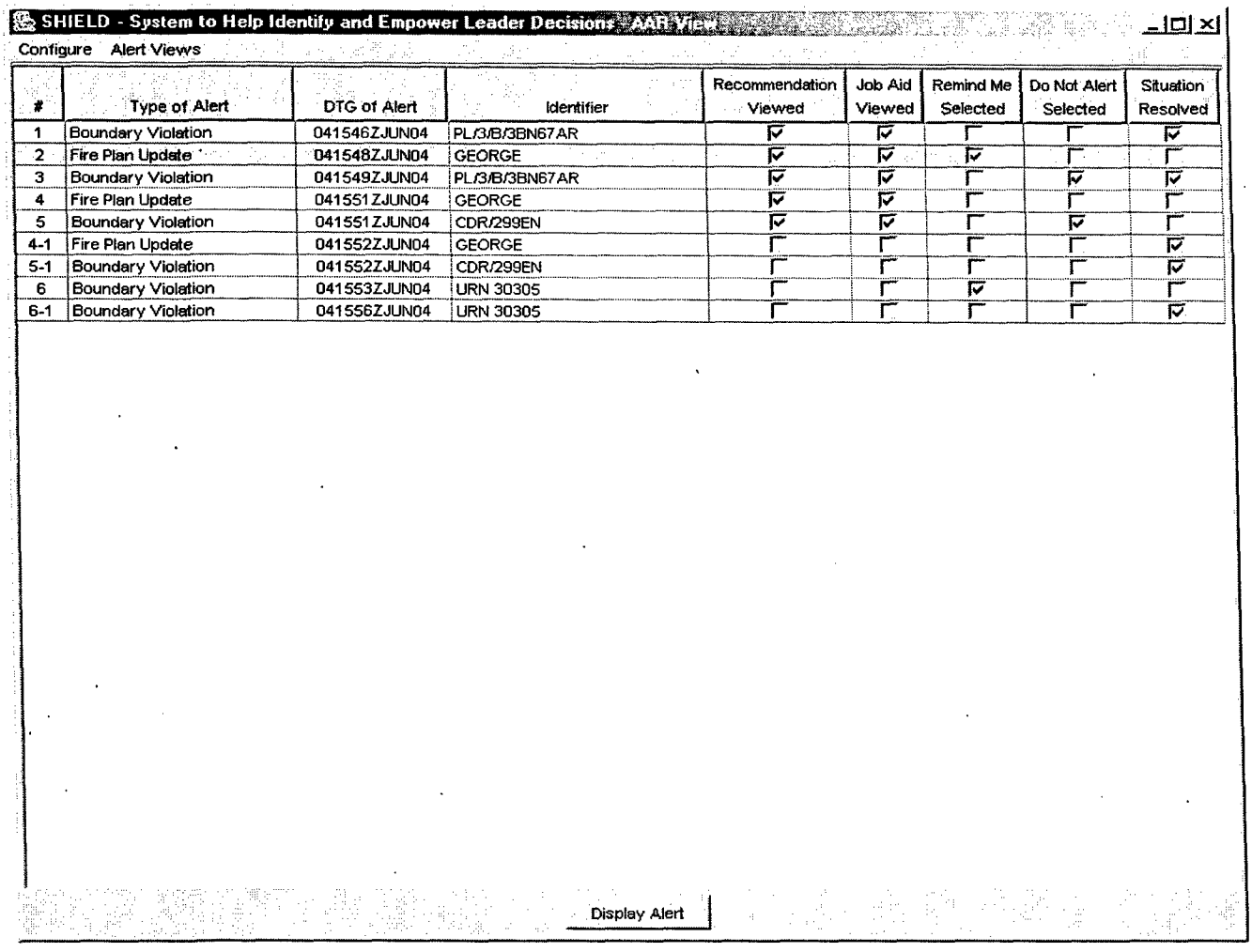


Figure 10. SHIELD and the training feedback process.

Interactive SHIELD AAR log file

After the operational event, SHIELD provides leaders with historical records of all the alerts, whether they monitored or dismissed the alerts, and whether they reviewed recommendations, job aids or both. The log also shows whether the events triggering alerts continue to persist during an exercise (i.e., the situation is not resolved). The log of alerts, actions and inactions permits leaders to select events they want to review and then display the alerts, recommendations, and job aids that they accessed or could have accessed during the exercise or real world operation for AAR purposes. Figure 11 illustrates a SHIELD AAR log.

AAR logs are available at a node where SHIELD is employed. For example, if SHIELD is employed on a company commander's FBCB2, the company commander can review his/her SHIELD AAR log at the end of an exercise. The commander can see how he/she responded to the various alerts received during mission. The commander might even call up the specific alerts of interest and view recommended courses of action and job aids. It is important to note that the leader at any SHIELD node has access to the SHIELD AAR log file during as well as after exercises. This means that if time becomes available during an exercise, a leader can check the status of any alert issues and/or initiate an AAR.



#	Type of Alert	DTG of Alert	Identifier	Recommendation Viewed	Job Aid Viewed	Remind Me Selected	Do Not Alert Selected	Situation Resolved
1	Boundary Violation	041546ZJUN04	PL/3/B/3BN67AR	☑	☑	☐	☐	☑
2	Fire Plan Update	041548ZJUN04	GEORGE	☑	☑	☑	☐	☐
3	Boundary Violation	041549ZJUN04	PL/3/B/3BN67AR	☑	☑	☐	☑	☑
4	Fire Plan Update	041551ZJUN04	GEORGE	☑	☑	☐	☐	☐
5	Boundary Violation	041551ZJUN04	CDR/299EN	☑	☑	☐	☑	☐
4-1	Fire Plan Update	041552ZJUN04	GEORGE	☐	☐	☐	☐	☑
5-1	Boundary Violation	041552ZJUN04	CDR/299EN	☐	☐	☐	☐	☑
6	Boundary Violation	041553ZJUN04	URN 30305	☐	☐	☑	☐	☐
6-1	Boundary Violation	041556ZJUN04	URN 30305	☐	☐	☐	☐	☑

Figure 11. SHIELD AAR Log.

The ability of SHIELD to keep track of multiple instances of an alert type was added through an iterative process. For example, SHIELD can discriminate between alerts upon the basis of the IDs of the platforms violating a boundary. Similarly, SHIELD can discriminate among different instances of mismatches between fire support plans and enemy locations.

It is important to point out that specific interactive SHIELD AAR log files are currently viewable only at the node where they are produced. Developing procedures for collecting and aggregating the information in these log files across nodes and exercises would support unit level AARs and research on the design and nodal placement of DAR aids. FSCX has just initiated an effort funded by ARI that includes implementing the capability to collect and aggregate SHIELD AAR logs across exercises and nodes. This capability will help the training and research communities envision how the log files might be employed in the context of a unit level AAR. The data aggregation capability will also have use in ARI research regarding the impacts of alerts on overall awareness of the tactical situation.

Beyond Alert-Based DAR Aids

The original intent of the SHIELD effort was to implement alerts that require specific corrective actions, as is true with the first two alerts implemented. For these alerts, there are specific measures available in deciding whether the action is taken (i.e., did the digital message traffic triggering the alert go away?). There may be other cases where an alert may be useful to a unit without being part of a situation that includes a specific corrective action. For example, units can improve situational awareness by manually inserting icons to show the location of non-digitized or non-reporting friendly elements. It may be important to alert units to the fact that these icons have been created to ensure everyone knows where the non-reporting friendly forces are located: however, unlike the situation with the first two alerts implemented, there is no reason to expect that the unit will do something to make the triggering C4I event (i.e., manually generated icons) go away. Even if manual icon generation is not needed as an alert, it may be useful to capture this event for AARs, because the use of manually generated friendly icons to ensure SA displays provide a more accurate picture of the friendly situation is an important indicator of a unit's ability to use FBCB2 to support operations (Liebrecht, Lockaby, and Meliza, 2003).

In addition to alerting units to threatening conditions that need to be addressed or to which units should give further attention, DAR aids can be used to make sure a unit is aware that certain conditions have been met (e.g., a unit has passed a phase line). This approach to using DAR aids does not assume that every condition potentially of interest can be addressed by a DAR aid. Instead it is assumed that enough of the conditions can be addressed to reduce the observation and analysis workloads of leaders to a significant extent. The practicality of implementing a greater variety of DAR aids depends to a large extent on where particular DAR aids are generated within a network, because of the potential for trying to create too many DAR aids at a node located on an existing C4I system and, interfering with the operation of that system. To preclude information overload from DAR aids, FBCB2 operators, for example, can take advantage of the SHIELD intrusiveness filter currently being developed to minimize the number of alerts they receive by activating just a select few alerts. A backup may be to have a SHIELD system or systems within the TOC track certain aspects of the tactical situation to the benefit of FBCB2 users. The tactical information loop is completed when an operator or leader in the TOC, or SHIELD system in the TOC responsible for the alert, relays the information to appropriate FBCB2 users. The intrusiveness filter and the TOC backup could be particularly beneficial for FBCB2 platform users who may be heavily engaged in mission execution.

It is also possible that SHIELD can be employed to create depictions of the tactical situation during exercises or missions for inclusion within the AAR log file without displaying these to leaders during the exercise/mission. Again, the practicality of such an approach depends upon whether it can be implemented without disrupting C4I system performance.

Under contract with ARI, FSCX is currently implementing new capabilities within SHIELD that will increase its ability to support research on the use of DAR aids, the utility of SHIELD-assisted AARs, and the impacts of SHIELD alerts on SA. This SHIELD enhancement includes six new DAR aids, two of which will be included in the AAR log file but not associated with an alert. Certain of the new alerts will differ from the previous alerts implemented in SHIELD in that they will not be associated with a triggering C4I event that a unit is expected to

remove through corrective actions. This newer version of SHIELD also includes the capability for users or researchers to selectively turn off the display of specific alerts. Finally, FSCX is preparing three canned exercise scenarios, counterbalanced in terms of the number of various types of alerts that can be triggered. Research questions that can be addressed using this newer version of SHIELD include:

- ✓ do leaders refer to the AAR log file to keep track of the tactical situation during exercises?
- ✓ does the availability of alerts have an impact on a user's overall awareness of the tactical situation?
- ✓ does the use of the SHIELD AAR log file for AARs have an impact on subsequent use of alerts and/or attention to specific aspects of the tactical situation?
- ✓ can SHIELD DAR aids and AAR logs be used to help train users to employ C4I displays to monitor the tactical situation, without causing negative training (i.e. can they be used in training without being available in an operational context?)?

C4I DATA STREAM TYPES AND ISSUES RELEVANT TO DAR AND AAR

There are a minimum of three types of C4I data streams that a system might draw upon to create DAR aids. There are the streams of C4I messages sent between or among different digital systems, the stream of C4I messages sent between the same system at different echelons, and C4I data internal to a C4I system.

C4I messages shared among different types of C4I systems

A substantial portion of the messages shared among C4I systems are in the form of USMTF and JVMF messages. During the development process for Prototype 1, FSCX learned a plan based on SHIELD reading USMTF messages to run the rule engine for a battalion/brigade-level TOC, was not valid. Nearly all message traffic at the battalion and brigade level uses JVMF messages and the devices in the lab used JVMF messages. This increased the complexity of the SHIELD project, particularly the analytical work, given the variable nature of the JVMF message formats versus the fixed nature of the USMTF messages. FSCX analyzed all 127 JVMF messages and identified the messages (or facts) on which rule sets would be based.

Many versions of JVMF messages have been produced. FSCX used JVMF Reissue 4 data for the SHIELD Phase II effort. Although SHIELD supports any of the JVMF versions, FSCX ended up using JVMF DCX2 in the lab so it would be compatible with the FBCB2 v.3.2.4.0 software used in the SHIELD simulation testbed.

In many cases, units may transmit information in the form of free text messages rather then using structured message formats. In such cases, SHIELD will have a difficult time gathering the information it needs to generate certain alerts. In addition, many planning products may be shared in the form of Microsoft PowerPoint and Word files. However, once units realize that SHIELD can provide them alerts to critical events by monitoring formatted messages, their motivation to use such formats should increase.

An important issue is whether it is cost-effective to develop an authoring tool that would allow units to develop their own SHIELD DAR aids. As mentioned in the second paragraph of this report, one of the corrective actions taken by units to improve future performance is to change TTPs to provide greater SA and understanding during missions. SHIELD with an authoring tool becomes a tool to effect TTP changes. One hundred twenty-seven JVMF messages and 376 USMTF messages were analyzed to identify facts that might be important in implementing rules for DAR aids. This effort identified over fifty macros (e.g., rules of engagement) or rule sets that might be used in implementing rules for generating DAR aids. Each macro requires from 2 to 18 JVMF or USMTF messages to provide the facts needed for a rule, and each of the messages contains from 10 to more than one hundred data fields. For example, the Rules of Engagement Macro would require analysis of fields within six JVMF messages (e.g., Commander's Fire Mission Guidance message) and nine USMTF messages (e.g., Target Bulletin message). Finding over fifty macros, many of which may be applied in developing multiple alerts, suggests a high payoff for developing a SHIELD rule authoring system. Although the large number of messages and message fields providing data for the macros suggests that developing an authoring tool may be a substantial effort, its potential for flexible and rapid rule development could be extremely valuable.

Message streams between or among the same system at different echelons

There are other structured digital messages beyond those in the JVMF and USMTF format that may be useful in preparing alerts. In particular there are unique system to system messages. For example, FSCX knew going into this effort that the current configuration of SHIELD would not monitor "native AFATDS" language – most of the message traffic sent from one AFATDS device directly to another AFATDS device. Although a couple of AFATDS to AFATDS messages are JVMF/USMTF based messages, most are not, including messages associated with fire plans. We began research to develop a parser that would parse the AFATDS to AFATDS traffic monitored by an AFATDS testing software product called Extensible C4I Instrumentation Suite (ExCIS). However, we discovered the tested version of ExCIS would not read AFATDS traffic associated with fire plans. A future version of ExCIS may permit resolution of this challenge.

Data internal to a particular C4I system

It would also be useful to collect and analyze certain information that is not transmitted among systems. For example, a product like SHIELD might be used to prepare AAR aids to determine if operators used system capabilities such as filter settings or analytical tools.

DAR DEVELOPMENT AND SITUATIONAL AWARENESS TESTBEDS

The requirements for this Phase II SBIR effort included the development of a testbed that could be used to conduct further research and development on the topic of DAR implementation issues, including the development of additional alerts. The identification of systems that could be used to stimulate SHIELD with C4I data streams was a high priority concern.

Iterative Development of a DAR Aid Development Testbed

The initial lab configuration we designed was based on research and agreements reached with government agencies during Phase I in order to establish a stable development and test environment that would be as near a tactical environment as possible. It included a combination of Government off-the-shelf (GOTS) software, Commercial off-the-shelf (COTS) software, and Government Furnished Equipment (GFE) hardware to be installed and configured. Upon award of Phase II (January 2003), however, we discovered that some of the software or hardware we needed for our simulation lab was either no longer available or its available version was not compatible with other simulation software. It required about four months to finally obtain all of the right software and hardware to make it work correctly as an operational simulation development environment. FSCX continued to evolve and refine the lab throughout the development of SHIELD to provide increased testing capabilities. Figure 12 below represents the development laboratory as of September 2003. Figure 13 represents the state of the lab as of March 2004. Figure 14 represents a variation of the lab configuration for testing with C2PC. See Appendix A for details of software and hardware versions.

The configuration described in Figure 12 was motivated by the desire to include a low cost driver for FBCB2 training. This low cost driver involved linking an inexpensive virtual simulation from the perspective of a vehicle commander into the testbed. This simulation had been used in conjunction with FBCB2 to create a situation where leaders could practice using FBCB2 in the context of simulated tactical missions. Reaching this goal required using the Situational Awareness Tactical Internet Data Server (SATIDS). FSCX found that the OneSAF Test Bed (OTB) through SATIDS configuration would only work for one of the rules being developed in Phase II (Cross Boundary-Fratricide Prevention) due to SATIDS limitations. In addition, FSCX was forced to upgrade SATIDS to support a more recent version of FBCB2, and the possibility of having to upgrade SATIDS again in response to future changes in FBCB2 made the original configuration too expensive to maintain and employ. A decision was made to switch from the development lab based on OTB in Figure 12 to one based on the Digital Battlestaff Sustainment Trainer-Low Overhead Driver (DBST-LOD) Joint Semi-Automated Force (JSAF) in Figure 13 to act as the scenario driver and translators in order to add AFATDS to the network and to be able to run other rule sets on SHIELD.

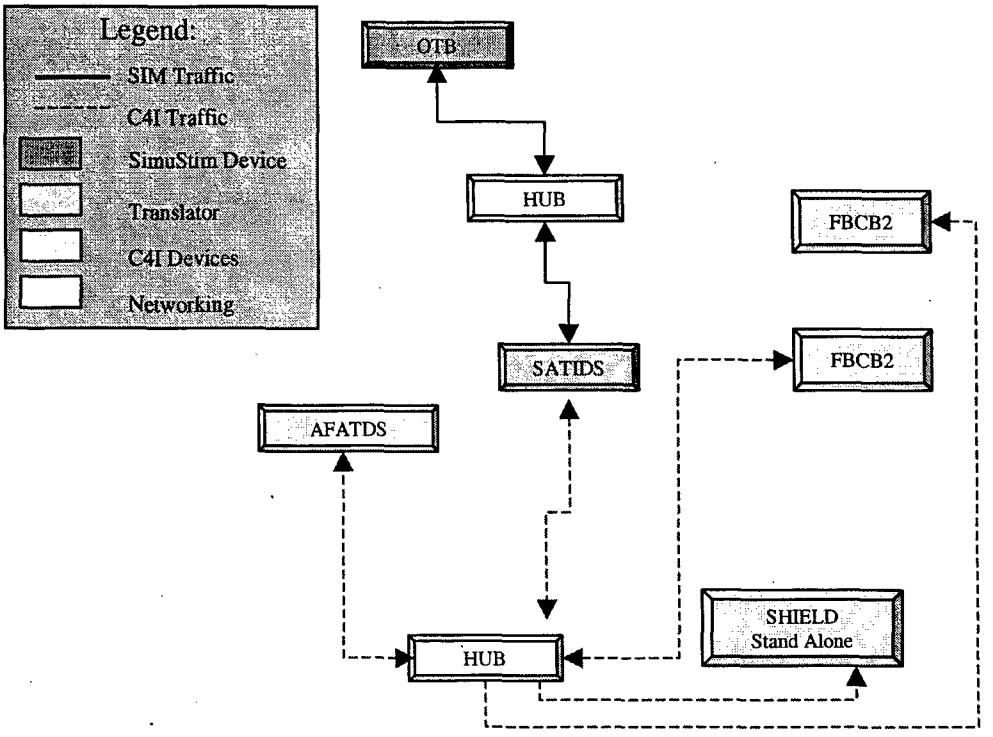


Figure 12. SHIELD development lab as of September 2003.

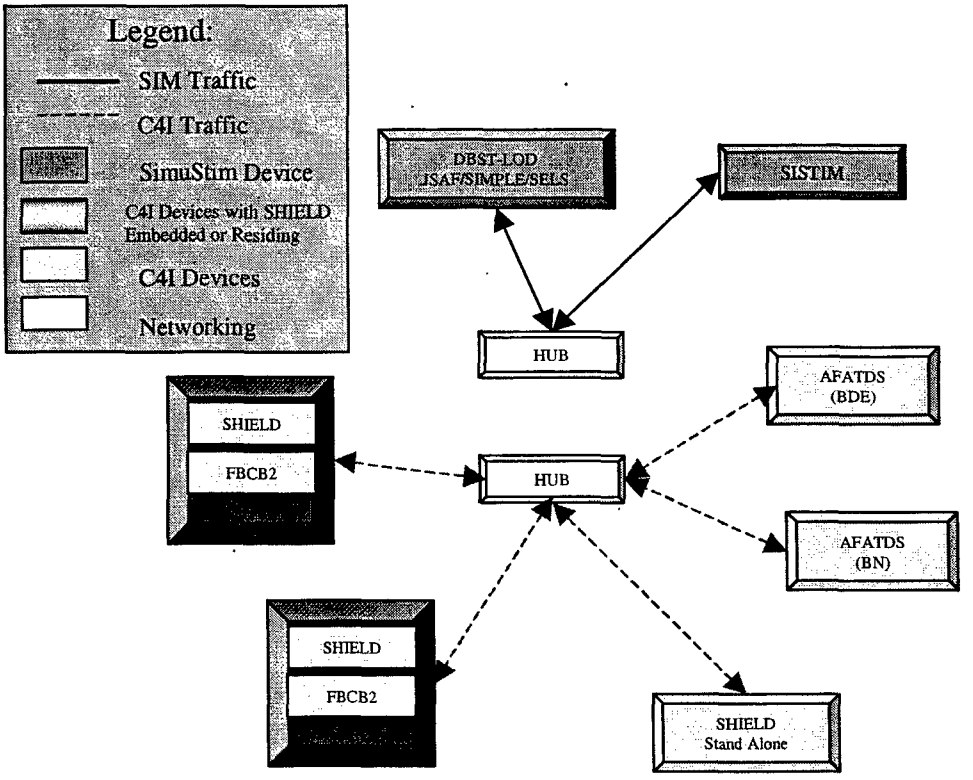


Figure 13. SHIELD development lab as of March 2004.

The C2PC lab configuration shown in Figure 14 provided FSCX with the capability of embedding SHIELD on a C4I system using C2PC APIs. It also provided another C4I system to demonstrate with SHIELD without changing the simulator/stimulator for the scenario.

The current lab environment enables the development and execution of exercise scenarios which trigger SHIELD rule sets. With this capability, we are not dependent on existing data-logged exercises from other locations that may or may not contain events and message traffic, which will initiate the generation of SHIELD alerts, enhanced SA displays, recommended courses of action, and job aids. Additionally, data-logged exercises from other sites may not be completely compatible with the software versions of the tactical digital systems in the FSCX lab, due to variations in the versions of C4I systems employed within various units.

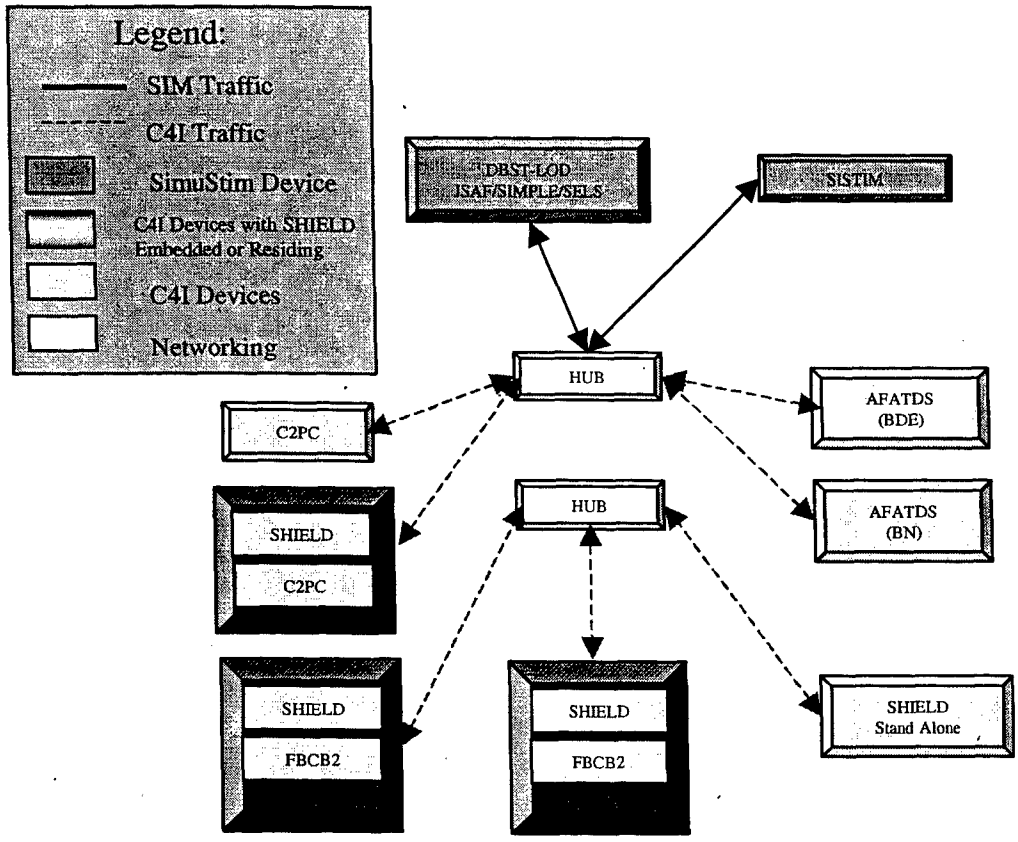


Figure 14. SHIELD development lab as of September 2004.

Situational Awareness Testbed

In an effort to simplify the process of demonstrating SHIELD to potential sponsors at distant sites, FSCX developed a process for loading and employing digital data stream scenarios on the same laptop computer used for the SHIELD and C4I system software. This capability also enables a wide variety of research issues to be addressed using a single laptop. As examples, research can be conducted to examine the impacts of alerts on overall SA, find out if

using alerts to train decision-makers to use SA displays leads to a dependence upon artificial cues that might not be available in combat, and find out if there are individual differences in terms of the willingness to employ alerts. FSCX is currently preparing three scenarios, counterbalanced in terms of the number and types of significant tactical events that can be observed, to support ARI SA and alert employment research.

SUMMARY

A series of SHIELD prototypes was developed under an OSD-funded SBIR Phase II project to demonstrate and explore the concept of a DAR system, capable of analyzing the C4I data stream to alert leaders to certain situations requiring their attention. Two alerts were implemented; one triggered when friendly units violate a boundary, another triggered when the location of planned artillery targets no longer corresponds to known enemy locations. Both alerts allow leaders to dismiss the alerts, have the system repeat the alert at a more opportune time, have the system refrain from repeating an alert for the rest of a mission, request recommended courses of action, and request job aids. As part of a larger effort, six additional rules – four alerts and two non-alerts - are being implemented and the SHIELD GUI is being modified to allow users to selectively turn specific alerts on or off at the start of a mission.

SHIELD logs alerts and user responses to these alerts as input for AARs. In the present version, the log file for a given node is available for use at that node. The decision-maker at each node can look at the responses to alerts (e.g., "I dismissed this alert three times before I called up a recommended course of action"), call up the alerts with associated enhanced SA displays, and look at the recommended courses of action and/or job aids. The user can also see whether the problem triggering the alert was addressed (i.e., was the mismatch between the fire plan and enemy locations corrected?).

Current efforts are directed towards collecting and aggregating information from log files across nodes and exercises as a step in applying these data to unit level AARs and research applications.

SHIELD prototypes have been demonstrated on a stand-alone hardware system, as an application running on two existing C4I systems (FBCB2 and C2PC), and as an application integrated with existing C4I software (C2PC). Instances of SHIELD alerts can be implemented at essentially any node within a network. SHIELD and its alerts were designed to be reused across C4I systems, allowing specific alerts to be moved from one node or nodes to another node or nodes where it will be more effective. SHIELD even collects the information regarding user responses to alerts needed to evaluate the effectiveness of node placement of specific alerts.

The overall goal of the SBIR topic to which the SHIELD project responded was to demonstrate the capability and value of implementing a system that can monitor C4I message traffic and alert decision-makers to critical situations in time to take corrective actions. The only non-negotiable requirement was that the information provided by the system should go beyond that which is easily provided by existing C4I systems. The approach taken in developing SHIELD demonstrated the capability and value of the subject system in the ways listed below.

- ✓ Alerts can be used to provide an enhanced awareness of the tactical situation, including the specifics regarding a situation to be addressed (e.g., identify and show the specific elements or unit violating a boundary) as opposed to leaving it up to a decision-makers to develop the specific information.
- ✓ Under certain situations, software can be used to decide if the situations triggering alerts have been addressed and document their status change, or lack thereof, for use during AARs.
- ✓ Alerts can be designed to reduce their intrusiveness.
- ✓ Recommended courses of action and job aids can be attached to the alerts to help decision-makers effectively respond, and they can be called up for review during AARs.
- ✓ Placement of specific alerts within a network can be can be adjusted to increase their effectiveness, and data collected by SHIELD can be used in assessing the effectiveness of differing placements.
- ✓ Software can be designed to support nodal flexibility in the placement of alerts.
- ✓ User responses to the alerts can be collected for use during AARs.
- ✓ The probability that alerting mechanisms will interfere with C4I system functions can be reduced through software design.
- ✓ The probability that alerting mechanisms will interfere with C4I system functions can be reduced by apportioning specific alerts among decision nodes within a network.
- ✓ An alerting mechanism that can be used at multiple nodes within a network concurrently offers the potential of supporting unit level AARs in addition to supporting decision node level AARs.
- ✓ There is the potential for using SHIELD capabilities to capture AAR aïds for situations other than alerts.

REFERENCES

Brown, B., Wilkinson, S., Nordyke, J., Riede, D., Huyssoon, S., Aguilar, D., Wonsewitz, R., & Meliza, L. L. (1997). *Developing an automated training analysis and feedback system for tank platoons* (ARI Research Report 1708). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. ADA 328445

Durlach, P.J. & Chen, J.Y.C. (2003). "Visual change detection in digital military displays." *Interservice/Industry Training, Simulation, and Education Conference 2003* Proceedings. Orlando, Fl.

Durlach, P.J. and Meliza, L.L. (2004). "The need for intelligent change alerting in complex monitoring and control systems." *Interaction between humans and autonomous systems over extended operation.* (AAAI Technical Report SS-04-03. 93-97)

Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1995). Design Patterns, Elements of Reusable Object-Oriented Software. Reading, MA: Addison-Wesley.

Leibrecht, B. C., Lockaby, K. J., & Meliza, L. L. (2003b). *A practical guide for exploiting FBCB2 capabilities* (ARI Research Product 2003-05). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. ADA 415997

Morrison, J. E. & Meliza, L. L. (1999). *Foundations of the after action review process* (Special Report 42). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. ADA 368651

APPENDIX A
SHIELD Laboratory – Configurations and Terminology

SHIELD Lab Configurations

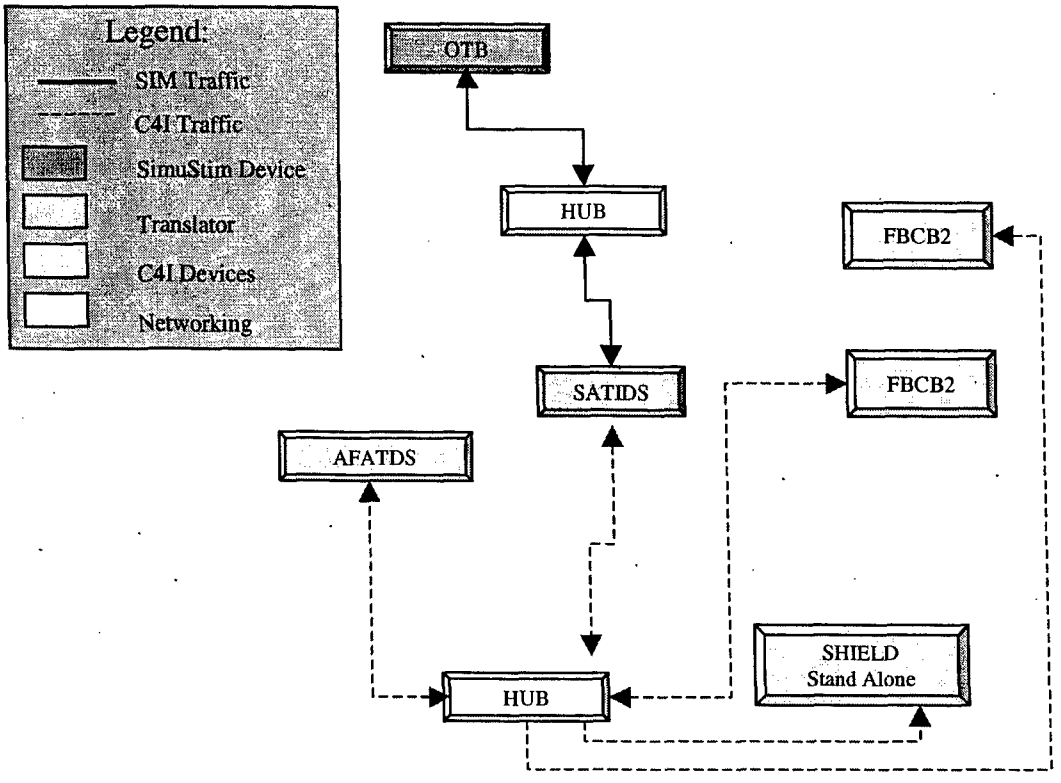


Figure 1. FSCX development lab as of September 2003.

Software	Version
AFATDS	v6.3.1
ETSIU	v3.4
FBCB2	v3.2.4.0
OTB	v2.0
SATIDS	v1.4.1
SHIELD	v1.1
Spearhead	Alpha release v1.1

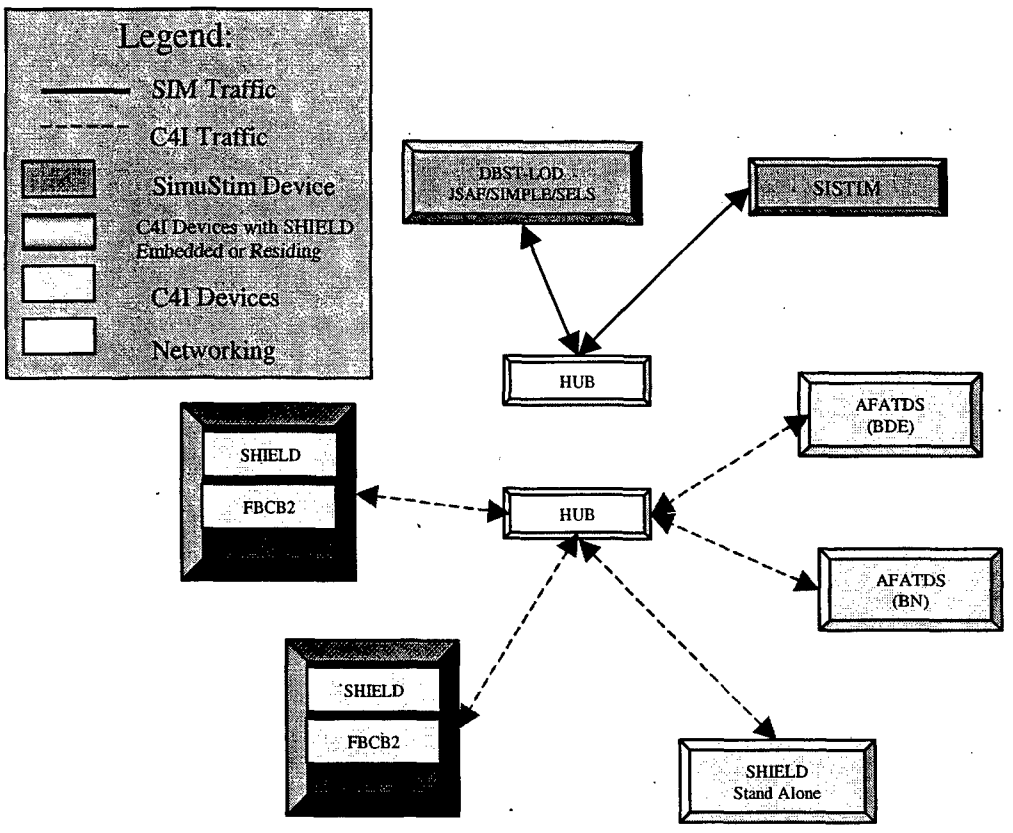


Figure 2. FSCX development lab as of March 2004.

Software	Version
AFATDS BDE	v6.3.2
AFATDS BN	v6.3.2
FBCB2	v3.5.4
DBST-LOD/JSAF	v1.1 (with SIMPLE/SELS)
SHIELD	v1.7
SISTIM	v6.3.2

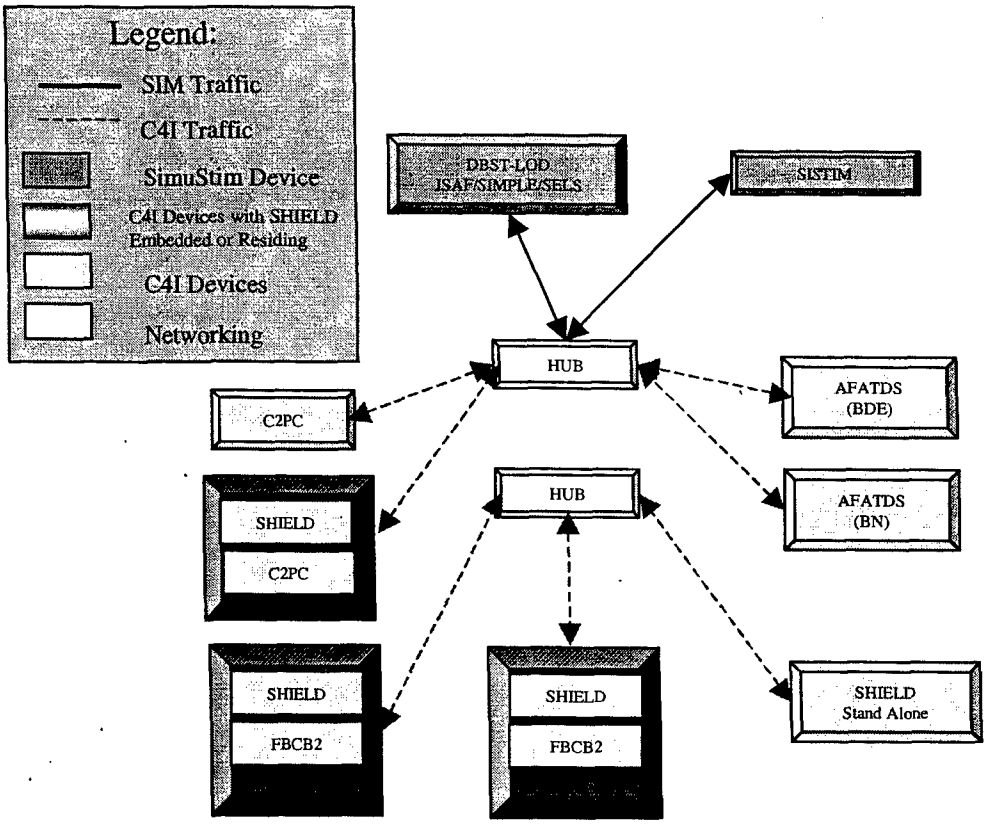


Figure 3. FSCX development lab as of September 2004.

Software	Version
AFATDS BDE	v6.3.2
AFATDS BN	v6.3.2
C2PC	v6.0.a4
FBCB2	v3.5.4
DBST-LOD/JSAF	v1.1 (with SIMPLE/SELS)
SHIELD	v1.7
SISTIM	v6.3.2

ACRONYMS

AAR	After action review
ABCS	Army Battle Command Systems (Army tactical level command, control, communications and computer systems)
AFATDS	Advanced Field Artillery Tactical Data System (ABCS system)
API	Application programming interfaces
ARI	US Army Research Institute for the Behavioral and Social Sciences
ASCII	American standard code for information exchange
BOS	Battlefield operating systems
C2PC	Command and Control Personal Computer (a Joint command, control, communications, computer and intelligence system)
C4I	Command, control, communications, computer, and intelligence
CCIR	Commander's Critical Information Requirements
CLIPS	C Language Integrated Production System
CLOS	Circular LOS
COE	Contemporary operational environment
COTS	Commercial off-the-shelf software
DAR	During After Action
DBST-LOD	Digital Battlestaff Sustainment Trainer-Low Overhead Driver
ETSIU	Enhanced Tactical Simulation Interface
ExCIS	Extensible C4I Instrumentation Suite
FBCB2	Force XXI Brigade and Below
GIF	Graphics interchange format
GFE	Government Furnished Equipment

GOTS	Government off-the-shelf
GUI	Graphical user interface
JCCB	Joint Configuration Control Board
JESS	Java Expert System Shell
JSAF	Joint Semi-Automated Force
JVMF	Format Joint Variable Message Text Format
LAN	Local area network
LOS	FBCB2 line-of-sight
MCS	Maneuver control system
OTB	One Semi Automated Forces Test Bed
RAM	Random access memory
SA	Situational awareness
SATIDS	Situational Awareness Tactical Internet Data Server (connects tactical C4I devices with simulations)
SBIR	Small business innovation research
SHIELD	System to Help Identify and Empower Leader Decisions
SIMPLE	Simulation C4I Interchange Module for Plans, Logistics and Exercises (C4I interface)
SELS	Scalable Entity Level Simulation (Artillery)
SISTIM	Simulation/Stimulation (provides simulation data to AFATDS and stimulates AFATDS)
SOP	Standard operating procedures
SSRU	Simulator Systems Research Unit
STO	Science and technology objective
TOC	Tactical operations centers

TRADOC	Training and Doctrine Command
TTP	Tactics, techniques and procedures
USMTF	United States message text format
XML	eXtensible markup language
XSLT	eXtensible stylesheet language transform

Spearhead – a virtual tank engagement simulation

List of Tables

List of Figures

with an enemy location). This SHIELD alert also shows enlarged icons showing known enemy locations and targets for a specific fire plan.

Figure 6. SHIELD geometry filter. The SHIELD geometry filter would allow the user to specify the area of operations in which data are to be monitored, the time that monitoring should begin or end, the name of an operations order to be monitored.

Figure 7. SHIELD intrusiveness filter. The intrusiveness filter would allow the user to turn off the display of all alerts and audio warnings in an all or none fashion. The filter would also allow the user to specify the time interval for repetitions of the same alert, select an option that no more than three alerts will be displayed at the same time, select an option that cause alert to be deleted if no action is taken within a period of time specified by the user, select an option that would cause alerts to be withheld if the user was in a user specified distance of enemy contact, and select an option that would alert the user when the user comes within a user specified distance to a decision point.

Figure 8. SHIELD rule set filter. This SHIELD filter would allow the user to select which alerts are to be active during an exercise or mission.

Figure 9. SHIELD-injected on C2PC. SHIELD alerts and/or a log of SHIELD alerts for the exercise are provided to the left of the C2PC situational awareness display.

Figure 10.SHIELD and the training feedback process. SHIELD maintains a log of the alerts generated and user responses to each alert. A trainer can review the log to identify trends or weaknesses.

Figure 11. SHIELD AAR log. This log contains a record of all alerts received by the user during a mission or exercise. Each alert is numbered. In cases where a specific alert situation is repeated, this fact is indicated by repeating the number and adding a dash with a second number indicating the number of the repetition (e.g., 1-2 if the same alert is displayed twice. The log also shows the users response to each alert, if any. Responses include viewing recommended courses of action, viewing job aids, having SHIELD repeat the alert at a later time, having the alert go away for the rest of the exercise, and/or resolving the situation.

Figure 12. SHIELD development lab as of September 2003. This development lab used OneSAF as the simulation and the Situational Awareness Tactical Internet Data Server to connect tactical C4I devices with simulations. One AFATDS and two FBCB2s were the C4I devices. This lab included SHIELD operating as a stand alone system.

Figure 13. SHIELD development lab as of March 2004. In this development lab DBST-Low Overhead Driver and SISTIM were used as the simulations. The C4I systems included one FBCB2 representing the platoon leader, one FBCB2 representing the platoon sergeant, one AFATDS representing battalion, and one AFATDS representing brigade. SHIELD operated as a stand alone system and as an application on two different FBCB2 systems.

Figure 14. SHIELD development lab as of September 2004. This version differs from the March 2004 version by adding SHIELD embedded on C2PC as an additional C4I device.